Networking from a network coding perspective

by

Tracey Ho

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2004

Author . . . .
Department of Electrical Engineering and Computer Science
May 24, 2004

Certified by . . . .
Muriel Médard
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Networking from a network coding perspective

by

Tracey Ho

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Network coding generalizes network operation beyond traditional routing, or store-and-forward, approaches, allowing for mathematical operations across data streams within a network. This thesis considers a number of theoretical and practical networking issues from network coding perspectives.

We describe a new distributed randomized coding approach to multi-source multicast network scenarios, in which one or more, possibly correlated, sources transmit common information to one or more receivers. This approach substantially widens the scope of applicability of network coding to three new areas. Firstly, we show that it achieves robust network operation in a decentralized fashion, approaching optimal capacity with error probability decreasing exponentially in the length of the codes. Secondly, in the area of network security, we show how to extend this approach to obtain a low-overhead scheme for detecting the presence of faulty or malicious nodes exhibiting Byzantine (arbitrary) behavior. Thirdly, we show that this approach compresses information where necessary in a network, giving error bounds in terms of network parameters.

Another area of our work develops an information theoretic framework for network management for recovery from non-ergodic link failures, based on the very general network coding concept of network behavior as a code. This provides a way to quantify essential management information as that needed to switch among different codes (behaviors) for different failure scenarios. We compare two different recovery approaches, and give bounds, many of which are tight, on management requirements for various network connection problems in terms of network parameters.

Thesis Supervisor: Muriel Médard
Title: Associate Professor

Acknowledgments

I would like to thank my adviser Muriel Médard for her invaluable guidance, encouragement and help in so many ways; my thesis committee members/collaborators Ralf Koetter, David Karger and Michelle Effros for being wonderful mentors also; my other collaborators Ben Leong and Jun Shi, with whom it has been a pleasure working; my family for their faithful support; and Yu-Han Chang, who has been my family away from home, and without whose help and support this thesis would still be very far from completion.

Contents

List of Tables

Chapter 1

Introduction

1.1 Network coding

Network coding generalizes network operation beyond traditional routing, or store-and-forward, approaches. Traditionally, coding is employed at source nodes for compression of redundant information or to provide protection against losses in the network; coding is also employed at the link level to protect against random errors or erasures on individual links. The network's usual task is to transport, unmodified, information supplied by source nodes.

Network coding, in contrast, treats information as mathematical entities that can be operated upon, rather than as unmodifiable objects to be transported. It allows interior network nodes to perform arbitrary operations on information from different incoming links. Its interest is in network-wide effects arising from coding across multiple links.

The first example highlighting the utility of network coding was given by Ahlswede et al. [1]. Figure 1-1 shows their famous example of a network for which coding in the interior of the network is necessary in order to achieve the maximum possible multicast transmission rate.

This example opens up a rich field of study. This thesis examines the utility of network coding beyond this type of application where coding is used to achieve network capacity. We show that network coding opens up powerful new ways to

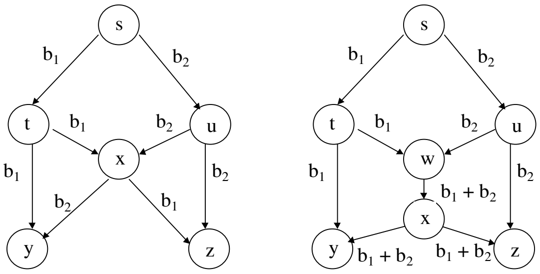


Figure 1-1: An example of a network requiring coding to achieve capacity (the network on the right, due to [1]) and a network which does not. Both networks consist of directed unit capacity links, and a source node s multicasting the same information to two receivers y and z. In the network on the left, no coding is required to multicast at a rate of 2 bits per unit time. In the network on the right, however, the presence of the bottleneck link from w to x necessitates coding on that link in order to achieve the same multicast rate. The labels on each link represent the information transmitted on that link in a scheme that achieves the maximum multicast rate of 2 bits per unit time.

consider and approach a number of operational and theoretical networking issues.

1.2 Contributions

One area of contribution is a new distributed randomized coding approach to multi-source multicasting, a rich family of networking scenarios which includes content distribution to multiple receivers, as well as the reachback problem for sensor networks in which multiple, possibly correlated, sources transmit to a receiver. This technique offers a means of achieving, in decentralized settings, the optimal capacity (given in [1]) achievable with centralized network coding.

In this approach, all nodes other than the receiver nodes perform random linear mappings from inputs onto outputs over some field. The mappings at each node are selected without coordination among nodes. To decode, the receivers need only know the overall linear combination of source processes in each of their incoming signals. Now it is relatively simple and inexpensive to communicate to the receivers the aggregate effect of the various random code choices in the network, rather than the individual random choices of every node. One possible method is to send this information with each signal block or packet as a vector of coefficients corresponding to each of the source processes, and have each coding node apply the same linear mappings to the coefficient vectors as to the information signals. The required overhead of transmitting these coefficients decreases with increasing length of blocks over which the codes and network state remain constant.

This distributed randomized approach substantially widens the scope of applicability of network coding to three new areas.

Firstly, it allows robust decentralized network operation, approaching optimal capacity with probability of unsuccessful transmission decreasing exponentially in the length of the codes. This offers a number of advantages. The ability to achieve, without coordination among nodes, the same transmission rates as with centralized planning, can be useful in settings where maintaining coordination is expensive or infeasible. Issues of stability, such as those arising from propagation of routing information, are obviated by the fact that each node selects its code independently from the others. The distributed nature of our approach also ties in well with considerations of robustness to changing network conditions. We show that our approach can take advantage of redundant network capacity for improved performance and robustness.

Secondly, in the area of network security, we show how to extend this approach to obtain a low-overhead scheme for detecting the presence of faulty or malicious nodes exhibiting Byzantine (arbitrary) behavior.

Thirdly, we show that a randomized network coding approach compresses information where necessary in a network, giving error bounds in terms of network parameters that generalize known error exponents for linear Slepian-Wolf coding [9] in a natural way.

Our results on distributed randomized network coding, more specifically, give a lower bound on the probability of error-free transmission for independent or linearly correlated sources, which, owing to the particular form of transfer matrix determinant polynomials, is tighter than the Schwartz-Zippel bound [50] for general polynomials of the same total degree. This bound, which is exponentially dependent on the code length, holds for any feasible set of multicast connections over any network topology (including networks with cycles and link delays). We further give, for acyclic networks, tighter bounds based on more specific network structure, and show the effects of redundancy and link reliability on success probability. We illustrate some possible applications with two examples of practical scenarios – distributed settings and online algorithms for networks with dynamically varying connections – in which randomized network coding shows promise of substantial benefits compared to routing-only approaches.

The addition of Byzantine modification detection capability is achieved by sending a simple polynomial hash value with each packet or block. The scheme requires only that a Byzantine attacker is unable to design and supply modified packets with complete knowledge of other packets received by other nodes. Additional computation is minimal as no cryptographic functions are involved. Detection probability can be traded off against communication overhead, field size (complexity) of the network code and the time taken to detect an attack.

For arbitrarily correlated sources, we consider the probability of decoding error using minimum entropy and maximum a posteriori probability decoding. We give, for two arbitrarily correlated sources in a general network, upper bounds on the probability of decoding error at a receiver, in terms of network parameters. In the special case of a Slepian-Wolf source network consisting of a link from each source to the receiver, our error exponents reduce to the corresponding results in [9] for linear Slepian-Wolf coding. The latter scenario may thus be considered a degenerate case of network coding.

Our analysis of randomized network coding uses relations we establish between multicast network coding and bipartite matching/network flows, leading to two alternative formulations of the algebraic condition of [39] for checking the validity of a linear network code. These new formulations draw precise relationships between network coding and network flows, and illuminate the mathematical structure of linear network codes. Besides their application in our analysis of randomized coding, these results have also led to a substantially tighter upper bound than previously known on the field size required for deterministic centralized network coding over general networks.

Another area of contribution is in developing a theory of network management that quantifies the management information needed fundamentally to direct recovery from link failures. With a high degree of generality, we can think of network behavior as being specified by a network code which gives the input-output relations of the network nodes. We can then quantify essential management information as that needed to switch among different codes (behaviors) for different failure scenarios. We compare two different recovery approaches, and provide bounds, many of which are tight, on management requirements for various network connection problems in terms of basic parameters such as the number of source processes and the number of links in a minimum source-receiver cut. Our results include a lower bound for arbitrary network connections and an upper bound for multi-transmitter multicast connections, for recovery using linear codes from all single link failures.

Parts of this work have appeared in [26], which introduced distributed randomized network coding, [25], which presented connections with bipartite matching/network flows and a new bound on required field size for centralized network coding, [31], which generalized previous results to arbitrary networks and gave tighter bounds for acyclic networks, [33] on the utility of network coding in dynamic environments, [32] on Byzantine modification detection, [27] on network coding for arbitrarily correlated sources, and [29, 28, 30] on network management.

1.3 Network coding background and related work

The field of network coding has its origins in the work of Ahlswede et al. [1] and Li et al. [43]. Ahlswede et al. [1] show that coding within a network allows a source to multicast information at a rate approaching the smallest minimum cut between the source and any receiver, as the coding symbol size approaches infinity. Their example, shown in Figure 1-1, demonstrates that this is not always possible without network coding. Li et al. [43] show that linear coding with finite symbol size is sufficient for multicast connections. Koetter and Médard [39] present an algebraic framework for linear network coding extending previous results to arbitrary networks and robust networking, and prove the achievability with time-invariant solutions of the min-cut max-flow bound for networks with delay and cycles. Reference [39] also gives an algebraic characterization of the feasibility of a multicast problem and the validity of a network coding solution in terms of transfer matrices, which we show in [25] has equivalent formulations related to bipartite matching and network flows. We use this result in obtaining a tighter upper bound on the required field size than the previous bound of [39], and in our analysis of distributed randomized network coding, introduced in [26]. Concurrent independent work by Sanders et al. [59] and Jaggi et al. [34] considers single-source multicast on acyclic delay-free graphs, showing a similar bound on field size by different means, and giving centralized deterministic and randomized polynomial-time algorithms for finding network coding solutions over a subgraph consisting of flow solutions to each receiver. A tighter field size bound for the case of two sources is given by Fragouli et al. [17], who consider network coding on two-source multicast networks as a graph coloring problem. Various practical protocols for and experimental demonstrations of randomized network coding [8] and non-randomized network coding [66, 52] have also been presented. Deb and Médard [10] present a gossip protocol using random linear coding.

A number of papers have considered the characteristics of network codes needed for achieving capacity on different types of networks and connections. Lower bounds on coding field size are presented by Rasala Lehman and Lehman [42] and Feder et al. [16]. Reference [16] also gives graph-specific upper bounds based on the number of "clashes" between flows from source to terminals. The need for vector coding solutions in some non-multicast problems is considered by Rasala Lehman and Lehman [42], Médard et al. [49] and Riis [57]. Reference [49] also gives a coding theorem that provides necessary and sufficient conditions, in terms of receiver entropies, for an arbitrary set of connections to be achievable on any network. Dougherty et al. show in [13] that linear coding is insufficient in general for non-multicast networks, and in [12] that the existence of a solution in some alphabet does not imply the existence of a solution in all larger non-finite field alphabets. Li and Li [44] consider undirected networks, showing that network coding does not increase throughput for a single unicast or broadcast session, while in the case of a single multicast session, any such increase in throughput is bounded by a factor of two.

Two other applications of network coding, error correction and protection against wiretapping, have been considered by Cai and Yeung. In [4] they give bounds on the sizes of information sources that can be transmitted with error-correcting network codes. In [6] they present a sufficient condition and a construction for network codes that prevent a wiretapper with access to a subset of links from obtaining information about any transmitted messages.

Another line of investigation taken by [45, 65] looks at coding over networks with link usage costs rather than link capacity constraints.

1.4 Thesis outline

In Chapter 2, we describe the basic algebraic model we use in our analyses. Chapter 3 establishes connections between network coding and bipartite matching/network flows. These connections are used in deriving a bound on required coding field size, as well as in the analysis of Chapter 4, which presents a distributed randomized network coding approach to multicasting in networks, focusing on the case of independent or linearly correlated sources. In Chapter 5 we show how to extend this randomized coding approach to detect Byzantine or faulty behavior in a network. The case of arbitrarily correlated sources is analyzed in Chapter 6. Chapter 7 considers requirements on network management information for recovery from failures. We present our conclusions and some directions for further work in Chapter 8.

Chapter 2

Basic model

This chapter sets out our basic model and mathematical framework, which is essentially based on that of [39]. The mathematical development of subsequent chapters builds on the analysis of this chapter, though we will introduce in some cases specializations, extensions or modifications of the model described here.

2.1 Network model

A network is represented as a directed graph with ν links (or edges) of unit capacity, i.e. 1 bit per unit time, and r independent discrete random processes $X_1, X_2, \ldots, X_r$, each of unit entropy rate, observable at one or more source nodes. General networks can be modeled to arbitrary accuracy by choosing a large enough time unit, and by representing edges with larger capacities as parallel edges and sources of larger entropy rate as multiple sources at the same node. We assume that all links have the same delay, modeling links with longer delay as multiple links in series. In the case that all links have zero delay, the network is considered *delay-free*; to ensure stability, such networks are assumed to be acyclic.

There are $d \geq 1$ receiver nodes. The output processes at a receiver node β are denoted $Z(\beta, i)$. A connection problem specifies, for each receiver, the subset of source processes to be transmitted to that receiver. A *multicast* connection problem is to transmit all the source processes to each of the receiver nodes.

Link l is an *incident outgoing link* of node v if $v = \text{tail}(l)$, and an *incident incoming link* of v if $v = \text{head}(l)$. We call an incident outgoing link of a source node a *source link* and an incident incoming link of a receiver node a *terminal link*. Links l_1 and l_2 are *incident* if $\text{head}(l_1) = \text{tail}(l_2)$ or $\text{head}(l_2) = \text{tail}(l_1)$. Edge l carries the random process $Y(l)$. A *path* is a subgraph of the network consisting of a sequence of links $e_1, \ldots, e_k$ such that e_i is an incident incoming link of e_{i+1}, and each node is visited at most once.

The random processes X_i, $Y(l)$, $Z(\beta, i)$ generate binary sequences. We assume that information is transmitted as vectors of bits. The length of the vectors is equal in all transmissions, and the symbol timing on all links is synchronized.

2.2 Coding model

We consider here linear coding[1]. In the basic scalar coding model, binary vectors of length u are viewed as scalar elements of the finite field $\mathbb{F}_{2^u}$. The random processes X_i, $Y(l)$, $Z(\beta, i)$ are thus represented as sequences $X_i = \{X_{i,0}, X_{i,1}, \ldots\}, Y(l) = \{Y_0(l), Y_1(l), \ldots\}, Z(\beta, i) = \{Z_0(\beta, i), Z_1(\beta, i), \ldots\}$ of symbols from $\mathbb{F}_{2^u}$.

Corresponding symbols from input and source sequences at a node are combined linearly in the field $\mathbb{F}_{2^u}$, i.e. the signal $Y(j)$ on a link j is a linear combination of processes X_i generated at node $v = \text{tail}(j)$ and signals $Y(l)$ on incident incoming links l. For the delay-free case, this is represented by the equation

$$Y(j) = \sum_{\{i \,:\, X_i \text{ generated at } v\}} a_{i,j} X_i + \sum_{\{l \;:\; \text{head}(l) \,=\, v\}} f_{l,j} Y(l)$$

and an output process $Z(\beta, i)$ at receiver node β is a linear combination of signals on its terminal links, represented as

$$Z(\beta, i) = \sum_{\{l \,:\, \text{head}(l) = \beta\}} b_{\beta_{i,l}} Y(l)$$

[1]which is shown in [43] to be sufficient for multicast.

For multicast on a network with link delays, memory is needed at the receiver nodes, but memoryless operation suffices at all other nodes [39]. The corresponding linear coding equations are

$$
\begin{aligned}
Y_{t+1}(j) &= \sum_{\{i\,:\,X_i \text{ generated at } v\}} a'_{i,j} X_{i,t} + \sum_{\{l\ :\ \text{head}(l)\,=\,v\}} f_{l,j} Y_t(l) \\
Z_{t+1}(\beta, i) &= \sum_{u=0}^{\mu} b'_{\beta_{i,l_u}} Z_{t-u}(\beta, i) + \sum_{\{l\,:\,\text{head}(l)=\beta\}} \sum_{u=0}^{\mu} b''_{\beta_{i,l_u}} Y_{t-u}(l)
\end{aligned}
$$

where μ represents the memory required. These equations, as with the random processes in the network, can be represented algebraically in terms of a delay variable D:

$$
\begin{aligned}
Y(j)(D) &= \sum_{\{i\,:\,X_i \text{ generated at } v\}} a_{i,j} X_i(D) + \sum_{\{l\ :\ \text{head}(l)\,=\,v\}} D f_{l,j} Y(l)(D) \\
Z(\beta, i)(D) &= \sum_{\{l\,:\,\text{head}(l)=\beta\}} b_{\beta_{i,l}} Y(l)(D)
\end{aligned}
$$

where

$$
\begin{aligned}
a_{i,j} &= D a'_{i,j} \\
b_{\beta_{i,l}} &= \frac{\sum_{u=0}^{\mu} D^{u+1} b''_{\beta_{i,l_u}}}{1 - \sum_{u=0}^{\mu} D^{u+1} b'_{\beta_{i,l_u}}}
\end{aligned}
$$

and

$$
\begin{aligned}
X_i(D) &= \sum_{t=0}^{\infty} X_{i,t} D^t \\
Y(j)(D) &= \sum_{t=0}^{\infty} Y_t(j) D^t, \qquad Y_0(j) = 0 \\
Z(\beta, i)(D) &= \sum_{t=0}^{\infty} Z_t(\beta, i) D^t, \qquad Z_0(\beta, i) = 0
\end{aligned}
$$

The coefficients $\{a_{i,j}, f_{l,j}, b_{\beta_{i,l}} \in \mathbb{F}_{2^u}\}$ can be collected into $r \times \nu$ matrices $A = (a_{i,j})$ and $B_\beta = (b_{\beta_{i,j}})$, and the $\nu \times \nu$ matrix $F = (f_{l,j})$, whose structure is constrained by

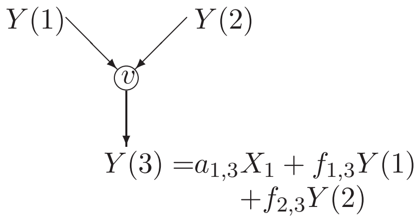


Figure 2-1: Illustration of linear coding at a node.

the network. Matrix A can be viewed as a transfer matrix from the source processes to signals on source nodes' outgoing links, and B as a transfer matrix from signals on terminal links to the output processes. F specifies how signals are transmitted between incident links. For acyclic graphs, we number the links ancestrally, i.e. lower-numbered links upstream of higher-numbered links, so matrix F is upper triangular with zeros on the diagonal. A triple (A, F, B), where

$$B = \left[\begin{array}{c} B_1 \\ \hline \vdots \\ \hline B_d \end{array} \right],$$

specifies the behavior of the network, and represents a *linear network code.*

The transfer matrix $M_\beta(A, F, B)$ describes the relationship between a vector of source values $[X_1\ X_2 \dots X_r]$ and the corresponding output vector $[Z_{\beta,1}\ Z_{\beta,2} \dots Z_{\beta,r}]$ at a receiver node β:

$$[X_1\ X_2 \dots X_r]\ M_\beta(A, F, B) = [Z_{\beta,1}\ Z_{\beta,2} \dots Z_{\beta,r}]$$

This transfer matrix $M_\beta(A, F, B)$ is given by AGB_β^T, where the matrix

$$G = \begin{cases} (I - F)^{-1} = I + F + F^2 + \dots & \text{in the acyclic delay-free case[2]} \\ (I - DF)^{-1} = I + DF + D^2F^2 + \dots & \text{in the case with delay[3]} \end{cases}$$

sums the gains along all paths in the network [39].

[2] The sequence converges since F is nilpotent for an acyclic network.
[3] The inverse exists since the determinant is a nonzero polynomial in D.

Linearly correlated sources can be brought into the same mathematical framework by considering unit source entropy and conditional entropy rates[4], and modeling such sources as pre-specified linear combinations of underlying independent unit entropy rate processes. Recovery of these underlying independent processes at the receivers means recovery of the correlated source processes as well. We denote these underlying independent processes by $X_1, X_2, \ldots X_r$ to match our earlier notation.

[4]As with independent sources, a large class of such sources can be modeled by choosing a large enough time unit and allowing for multiple sources at the same node.

Chapter 3

Connections with Bipartite Matching and Network Flows

In this chapter, we consider, from perspectives related to bipartite matching and network flows, network coding for independent or linearly correlated sources over arbitrary networks that may have cycles and link delays.

3.1 Main results and discussion

Reference [39] gives the following necessary and sufficient condition for a multicast connection problem with independent or linearly correlated sources to be feasible (or for a particular network code (A, F, B) to be a valid solution): that for each receiver β, the transfer matrix

$$M_\beta(A, F, B) = \begin{cases} A(I-F)^{-1}B_\beta^T & \text{in the acyclic delay-free case} \\ A(I-DF)^{-1}B_\beta^T & \text{in the case with delays} \end{cases}$$

mapping a vector of input values to the corresponding output vector at β has nonzero determinant.

The following result, which we prove in Section 3.2, is an alternative formulation of this condition that makes a connection with the Edmonds matrix [50] of bipartite matching, and can be used to easily deduce various characteristics of the transfer matrix determinant. We use the notation of the previous section, where r is the number of source processes, and ν is the number of links in the network.

Theorem 1 *(a) For an acyclic delay-free network, the determinant of the transfer matrix $M_1 = A(I-F)^{-1}B_\beta^T$ for receiver β in a network code (A, F, B) is equal to*

$$|M_1| = (-1)^{r(\nu+1)} |M_2|$$

where $M_2 = \begin{bmatrix} A & 0 \\ I-F & B_\beta^T \end{bmatrix}$ is the corresponding Edmonds matrix.
(b) For an arbitrary (possibly cyclic) network with unit delay links, the transfer matrix $A(I-DF)^{-1}B_\beta^T$ for receiver β in a network code (A, F, B) is nonsingular if and only if the corresponding Edmonds matrix $\begin{bmatrix} A & 0 \\ I-DF & B_\beta^T \end{bmatrix}$ is nonsingular. $\square$

Theorem 1 shows the equivalence of the network coding transfer matrix formulation and the Edmonds matrix formulation for checking if a bipartite graph has a perfect matching, which is a classical reduction, illustrated in Figure 3-1, of the problem of checking the feasibility of an $s-t$ flow [35]. This latter problem is a special case of network coding, restricted to the binary field and to separate transmission of different signals; it is interesting to find that the two formulations are equivalent for the more general case of coding in higher order fields.

The combinatorial formulations of Theorem 1 and Theorem 3 below connect network coding with network flows, providing more direct insights into how individual code coefficients affect the overall network code, and making it easier to deduce various characteristics of transfer matrix determinant polynomials without the complication of dealing with matrix products and inversions. For instance, Theorem 1 sheds light on the maximum exponent of a variable, the total degree of the polynomial, and its form for networks with linearly correlated sources.

These new insights into the structure of the transfer matrix determinant allow us to obtain two main results. One of them is Theorem 4 of Chapter 4, which uses Theorem 1 to deduce the total degree of the transfer matrix determinant polynomial

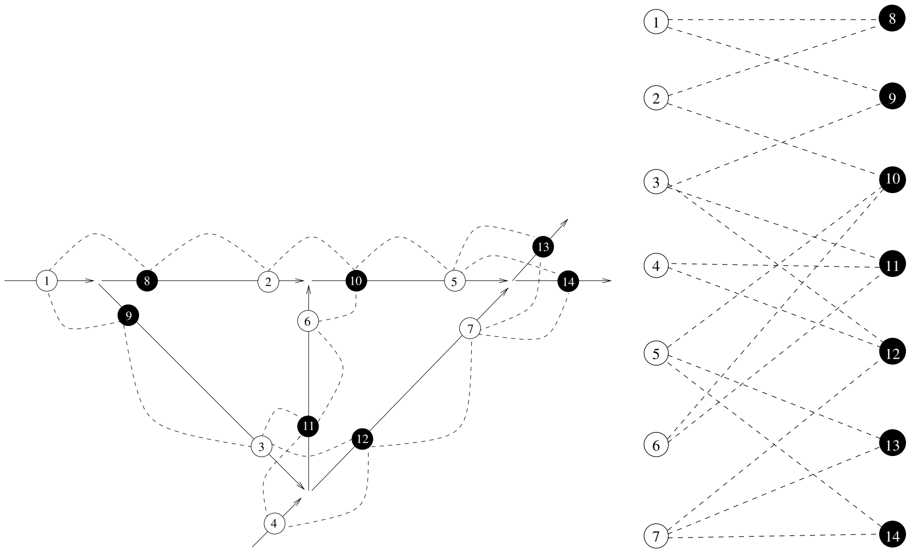


Figure 3-1: Example illustrating the reduction of a flow feasibility problem to a bipartite matching problem. Each link in the flow feasibility problem on the left is associated with one white bipartite node and one black bipartite node. Each source process is associated with one white bipartite node (nodes 1 and 4 in this example), and each output process with one black node (nodes 13 and 14 in this example). Connections between bipartite nodes (shown by dashed lines) are made between opposite-colored nodes of adjacent links and source/output processes.

for each receiver as well as the maximum exponent of each random coefficient, giving a bound on randomized coding success rate that is tighter than the Schwartz-Zippel bound for general polynomials of the same total degree. The other is a new upper bound on required field size for a feasible network coding problem:

Theorem 2 *For a feasible multi-source multicast connection problem with independent or linearly correlated*[1] *sources on an arbitrary network (which may have cycles and delay), there exists a solution in any finite field* $\mathbb{F}_q$ *where* q *is greater than the number of receivers* d. □

This substantially tightens the upper bound of $q > rd$ given in [39], where r is the number of processes being transmitted in the network. References [34, 59] independently and concurrently showed the sufficiency of $\mathbb{F}_q, q \geq d$ for the acyclic delay-free case.

[1]Section 2.2 describes our model of linearly correlated sources

For acyclic networks, Theorem 3 can be used in place of Theorem 1 to deduce many of the same transfer matrix determinant properties. Theorem 3 further allows us to tighten, for acyclic networks, the bound on randomized coding success probability in Theorem 4. This is used in our analysis of randomized coding on grid networks in Section 4.6.

Theorem 3 *A multicast connection problem is feasible (or a particular (A, F) can be part of a valid solution) if and only if each receiver β has a set $\mathcal{H}_\beta$ of r incident incoming links $h_1, \ldots, h_r$ for which*

$$\sum_{\substack{\{\text{disjoint paths } \mathcal{E}_1, \ldots, \mathcal{E}_r : \\ \mathcal{E}_i \text{ from outgoing link} \\ l_i \text{ of source } i \text{ to } h_i \in \mathcal{H}_\beta\}}} \left|A_{\{l_1,\ldots,l_r\}}\right| \prod_{j=1}^{r} g(\mathcal{E}_j) \neq 0$$

where $A_{\{l_1,\ldots,l_r\}}$ is the submatrix of A consisting of columns corresponding to links $\{l_1, \ldots, l_r\}$, and

$$g(\mathcal{E}) = \begin{cases} f_{e_1,e_2} f_{e_2,e_3} \cdots f_{e_{k-1},e_k} & \text{if } k > 1 \\ 1 & \text{if } k = 1 \end{cases}$$

is the product of gains on the path $\mathcal{E}$ consisting of links $e_1 < \ldots < e_k$. The sum is over all flow solutions that transmit all source processes to links in $\mathcal{H}_\beta$, each such solution being a set of r disjoint paths each connecting a different source to a different link in $\mathcal{H}_\beta$. $\square$

3.2 Proofs and ancillary results

The following proof is due to Jun Shi. Our original longer proof is given in Appendix A.

Proof of Theorem 1:

(a) Note that

$$\begin{bmatrix} I & -A(I-F)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & 0 \\ I-F & B_\beta^T \end{bmatrix} = \begin{bmatrix} 0 & -A(I-F)^{-1}B_\beta^T \\ I-F & B_\beta^T \end{bmatrix}$$

The first matrix, $\begin{bmatrix} I & -A(I-F)^{-1} \\ 0 & I \end{bmatrix}$, has determinant 1. So $\det\left(\begin{bmatrix} A & 0 \\ I-F & B_\beta^T \end{bmatrix}\right)$ equals $\det\left(\begin{bmatrix} 0 & -A(I-F)^{-1}B_\beta^T \\ I-F & B_\beta^T \end{bmatrix}\right)$, which can be expanded as follows:

$$\begin{aligned}
&\det\left(\begin{bmatrix} 0 & -A(I-F)^{-1}B_\beta^T \\ I-F & B_\beta^T \end{bmatrix}\right) \\
&= (-1)^{r\nu}\det\left(\begin{bmatrix} -A(I-F)^{-1}B_\beta^T & 0 \\ B_\beta^T & I-F \end{bmatrix}\right) \\
&= (-1)^{r\nu}\det(-A(I-F)^{-1}B_\beta^T)\det(I-F) \\
&= (-1)^{r(\nu+1)}\det(A(I-F)^{-1}B_\beta^T)\det(I-F)
\end{aligned}$$

The result follows from observing that $\det(I-F)=1$.

(b) By similar manipulations, we can show that

$$\det\left(\begin{bmatrix} A & 0 \\ I-DF & B_\beta^T \end{bmatrix}\right) = (-1)^{r(\nu+1)}\det(A(I-DF)^{-1}B_\beta^T)\det(I-DF)$$

Since $\det(I-DF)$ is nonzero, the result follows. ■

Proof of Theorem 2: By Theorem 1, the form of the transfer matrix determinant $|AGB_\beta^T|$ for any receiver β matches the form of the determinant of the Edmonds matrix, in which no variable $a_{x,j}$, $f_{i,j}$ or $b_{\beta_{i,j}}$ appears in more than one entry. Thus, no product term in the determinant polynomial for any receiver contains a variable $a_{x,j}$, $f_{i,j}$ or $b_{\beta_{i,j}}$ raised to an exponent greater than 1, and the largest exponent of any variable in the product P of d receivers' determinant polynomials is at most d.

We use an induction argument similar to that in [39] to show that there exists a solution in $\mathbb{F}_q, q > d$, such that P is nonzero. Consider one of the variables, denoting it by ξ_1, and consider P as a polynomial in the other variables (and D in the case with delays), with coefficients from $\mathbb{F}_2[\xi_1]$. Since these coefficients have maximum degree d, they are not divisible by $\xi_1^q - \xi_1$. Thus, ξ_1 can take some value in $\mathbb{F}_q$ such that at least one of the coefficients is nonzero. Repeating this procedure for each of the other variables gives the desired result. ■

Proof of Theorem 3: Recall that we assume an ancestral numbering for the links of an acyclic graph, i.e. lower-numbered links upstream of higher-numbered links. For $1 \leq h' \leq h \leq \nu$, let $S_{h',h}$ be the set of all sets of integers $\{e_1, e_2, \ldots, e_k\}$ such that $h' = e_1 < e_2 < \ldots < e_k = h$. Let $\mathcal{H} = \{h_1, \ldots, h_r\}$, where $1 \leq h_1 < \ldots < h_r \leq \nu$.

Let $\underline{a}_j$ and $\underline{c}_j$ denote column j of A and AG respectively, and let $G_{\mathcal{H}}$ denote the submatrix consisting of columns of G corresponding to links in set $\mathcal{H}$. It follows from the definitions of transfer matrices A and $G = I + F + F^2 + \ldots$ that $\underline{c}_h, h = 1, \ldots, \nu$, can be computed recursively as follows:

$$\underline{c}_1 = \underline{a}_1 \tag{3.1}$$

$$\underline{c}_h = \sum_{i=1}^{h-1} \underline{c}_i f_{i,h} + \underline{a}_h, \quad h = 2, 3, \ldots, \nu \tag{3.2}$$

Intuitively, $\underline{c}_h$ represents the overall response on link h to the source signals, calculated as the superposition of responses of upstream links i weighted by the corresponding gains $f_{i,h}$. Carrying out the recursive computation gives

$$\underline{c}_h = \sum_{i=1}^{h} \underline{a}_i \sum_{\mathcal{E} \in S_{i,h}} g(\mathcal{E}).$$

Using this expression for each column of $AG_{\mathcal{H}}$ and expanding the determinant linearly in all columns, we obtain

$$AG_{\mathcal{H}}| = \left| \begin{matrix} | & & | \\ \underline{c}_{h_1} & \ldots & \underline{c}_{h_r} \\ | & & | \end{matrix} \right|$$

$$= \sum_{\substack{\{(h'_1,\ldots,h'_r):\\ 1\le h'_j\le h_j\\ h'_i\ne h'_j\ \forall\, i\ne j\}}} \begin{vmatrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{vmatrix} \prod_{i=1}^{r} \sum_{\mathcal{E}\in S_{h'_i,h_i}} g(\mathcal{E})$$

$$= \sum_{\substack{\{(h'_1,\ldots,h'_r):\\ 1\le h'_j\le h_j\\ h'_i\ne h'_j\ \forall\, i\ne j\}}} \begin{vmatrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{vmatrix} \sum_{\substack{\{(\mathcal{E}_1,\ldots,\mathcal{E}_r):\\ \mathcal{E}_j\in S_{h'_j,h_j}\}}} \prod_{j=1}^{r} g(\mathcal{E}_j).$$

The above expansion does not take into account dependencies among the columns $\underline{c}_h$. We can obtain an equivalent expression with fewer terms by using the following alternative sequence of expansions which takes the dependencies into account. We start by expanding the determinant of $AG_{\mathcal{H}}$ linearly in the $h_r{}^{th}$ column using (3.2):

$$\begin{aligned} |AG_{\mathcal{H}}| &= \begin{vmatrix} | & & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_r} \\ | & & | \end{vmatrix} \\ &= \sum_{\substack{\{i\ :\ 1\le i< h_r,\\ i\ne h_1,\ldots,h_{r-1}\}}} \begin{vmatrix} | & & | & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_{r-1}} & \underline{c}_i \\ | & & | & | \end{vmatrix} f_{i,h_r} + \begin{vmatrix} | & & | & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_{r-1}} & \underline{a}_{h_r} \\ | & & | & | \end{vmatrix} \end{aligned}$$

and proceed recursively, expanding each determinant linearly in its column $\underline{c}_h$ whose index h is highest, using (3.2) for $h > 1$ and (3.1) for $h = 1$. At each expansion stage, the expression for $AG_{\mathcal{H}}$ is a linear combination of matrix determinants. Each nonzero determinant corresponds to a matrix composed of columns $\{\underline{a}_{k_1},\ldots,\underline{a}_{k_s},\underline{c}_{k_{s+1}},\ldots,\underline{c}_{k_r}\}$ such that $k_i \ne k_j\ \forall\, i \ne j$, and $\min(k_1,\ldots,k_s) > \max(k_{s+1},\ldots,k_r)$. Its coefficient in the linear combination is a product of terms $f_{i,h}$ such that $h > k_{s+1},\ldots,k_r$, and is of the form $\prod_{j=1}^{r} g(\mathcal{E}_j)$ where $\mathcal{E}_j \in S_{k_{j'},h_j}$ and $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset\ \forall\, i \ne j$. By induction we have that these properties hold for all nonzero determinant terms in the course of the expansion. The expansion terminates when the expression is a linear combination of determinants of the form $\left|\underline{a}_{l_1} \dots \underline{a}_{l_r}\right|$, at which point we have

$$|AG_{\mathcal{H}}| = \sum_{\substack{\{(h'_1,\dots,h'_r):\\ 1 \le h'_j \le h_j,\\ h'_i \ne h'_j \ \forall\, i \ne j\}}} \left| \begin{matrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{matrix} \right| \sum_{\substack{\{(\mathcal{E}_1,\dots,\mathcal{E}_r):\\ \mathcal{E}_j \in S_{h'_j,h_j},\\ \mathcal{E}_i \cap \mathcal{E}_j = \emptyset\\ \forall\, i \ne j\}}} \prod_{j=1}^{r} g(\mathcal{E}_j).$$

The result follows by noting that each set $\mathcal{E} = \{e_1, e_2, \dots, e_k\}$ such that $g(\mathcal{E}) \ne 0$ corresponds to a network path consisting of links $e_1, \dots, e_k$; that the condition $\mathcal{E}_j \cap \mathcal{E}_k = \emptyset$ for all $j \ne k,\ 1 \le j, k \le r$ implies that the corresponding paths $\mathcal{E}_1, \dots, \mathcal{E}_r$ are disjoint; and that $\left|\underline{a}_{h'_1} \dots \underline{a}_{h'_r}\right|$ is nonzero only when links $h_{j'}$ are source links carrying r independent signals. ■

Chapter 4

Distributed randomized network coding

Building on the mathematical groundwork of the previous chapter, we proceed to describe and analyze a distributed randomized network coding approach to transmitting and compressing information in networks. In this chapter we consider multicasting from one or more independent or linearly correlated sources over arbitrary directed networks which may have cycles and link delays. We will consider the case of arbitrarily correlated sources in Chapter 6.

4.1 Model and approach

Our approach is based on the scalar finite field coding framework described in Chapter 2. A sequence of bits generated at a source or transmitted on a link is grouped into strings of equal length u, each representing a symbol in the finite field $\mathbb{F}_{2^u}$. Nodes other than the receiver nodes independently and randomly select scalar linear mappings from inputs to outputs in the field $\mathbb{F}_{2^u}$, i.e. values for the code coefficients $\{a_{i,j}, f_{l,j}\}$ defined in Section 2.2 are selected uniformly at random from field $\mathbb{F}_{2^u}$. An illustration is given in Figure 4.1.

The randomized network coding approach allows for some or all of the coefficients $\{a_{i,j}, f_{l,j}\}$ to be chosen randomly, as long as the fixed coefficient values preserve

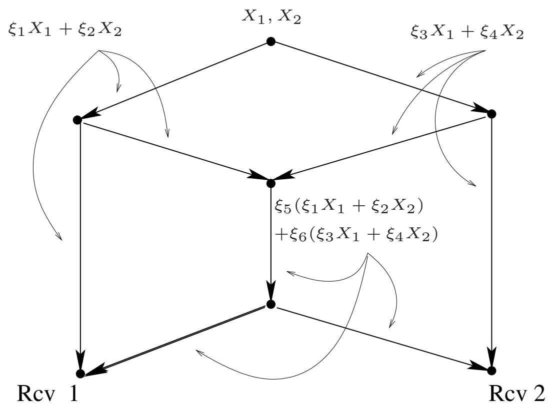


Figure 4-1: An example of distributed randomized network coding. X_1 and X_2 are the source processes, represented as sequences of symbols from a finite field, being multicast to the receivers, and the coefficients ξ_i are randomly chosen elements of the finite field. The label on each link represents the signal being carried on the link.

feasibility. We denote by η the number of links j with associated random coefficients $\{a_{i,j}, f_{l,j}\}$.

The aggregate effect of the random network coding must be communicated to the receivers for each block of symbols that is coded differently. Such blocks could correspond to different packets, or to longer sequences in relatively stable networks. One simple approach is to send, with each block, a vector of finite field coefficients each corresponding to a source process, and have each coding node apply the same linear mappings to the coefficient vectors as to the data sequences. One could also consider "sounding" the network by sending a canonical basis from the sources, an approach akin to measuring an "impulse response".

A random network code is valid, or successful, if all receivers are able to reconstruct the source processes. We use the conditions of Chapter 3 for checking the validity of a network code.

4.2 Main results

Theorem 4 *For a feasible multicast connection problem on an arbitrary network with independent or linearly correlated*[1] *sources, and a network code in which some or all code coefficients are chosen independently and uniformly over all elements of a finite field $\mathbb{F}_q$ (some coefficients can take fixed values as long as these values preserve feasibility*[2]*), the probability that all the receivers can decode the source processes is at least $(1 - d/q)^\eta$ for $q > d$, where d is the number of receivers and η is the number of links with associated randomized coefficients.* □

The complexity of the code grows as the logarithm of the field size $q = 2^u$, since arithmetic operations are performed on codewords of length u. The error bound is on the order of the inverse of the field size, so the error probability decreases exponentially with the number of codeword bits u.

The bound of Theorem 4 is very general, applying across all networks with the same number of receivers and the same number of links with independently chosen random linear mappings. Our next goal is to find tighter bounds by taking into account more specific network characteristics. One such bound, for acyclic networks with or without link delays, is based on a connection between randomized coding success probability and network connection feasibility when links are independently deleted with some probability.

Theorem 5 *For a d-receiver multicast problem on an acyclic network with independent or linearly correlated sources, the success probability of a random network code in the field of size q is greater than or equal to the probability that the network connections remain feasible after deleting each link of the original graph with probability d/q.*

The above bound is useful in cases where analysis of connection feasibility is easier than direct analysis of randomized coding. We apply it to obtain the following result

[1]Section 2.2 describes our model of linearly correlated sources

[2]i.e. the result holds for networks where not all nodes perform random coding

showing how spare network capacity and/or more reliable links allow us to use a smaller field size to surpass a particular success probability.

Theorem 6 *For a multicast problem on an acyclic network with independent or linearly correlated sources of joint entropy rate r, and links which fail (are deleted from the network) with probability p, let y be the minimum redundancy, i.e. deletion of any y links in the network preserves feasibility. A random network code transmits all source processes successfully to a particular receiver with probability at least*

$$\sum_{x=r}^{r+y} \binom{r+y}{x} \left(1 - p - \frac{1-p}{q}\right)^{Lx} \left(1 - \left(1 - p - \frac{1-p}{q}\right)^{L}\right)^{r+y-x}$$

where L is the longest source-receiver path in the network.

We motivate our interest in randomized network coding with two examples of practical advantages offered by this approach. The first is in distributed operation in environments where coordination is infeasible or expensive. We consider communication on a grid network as a simple example, obtaining an analytical upper bound on the performance of a distributed randomized flooding approach, which is exceeded by our lower bound on the performance of randomized coding for modest code lengths.

The second is in online operation in dynamic environments. As an illustration, we compare, for dynamically varying connections on randomly generated graphs, distributed randomized coding with an approximate online Steiner tree routing approach from [37] in which, for each transmitter, a tree is selected in a centralized fashion. The high complexity of such a routing scheme requires a simulation-based approach. In practice, an alternative to growing the field size (code length) for improving randomized coding success probability is to allow retrial of random codes in case of failure. We find that for randomly generated graphs of 8 to 10 nodes, randomized coding with 4-5 bit code lengths and a limit of 3 re-tries per new connection generally performs as well as, and in a non-negligible set of cases, better than the approximate Steiner-tree routing scheme.

Details and proofs of these results are given in the following sections.

4.3 Delay-free Networks

We first analyze delay-free networks, which must be acyclic in order for all signals to be well-defined. The analysis and results of this section apply also to acyclic networks with delay that are operated in a burst-oriented [43], pipelined [59] or batch-like [8] fashion, where information may be buffered or delayed so as to be combined with other incoming information from the same batch. A cyclic graph with v nodes and rate r may also be converted to an expanded acyclic graph with κv nodes and rate at least $(\kappa - v)r$, communication on which can be emulated over κ time steps on the original cyclic graph [1].

Lemma 1 *Consider a random network code (A, F, B) in which η links have associated randomized coefficients. The determinant polynomial of the corresponding Edmonds matrix* $\begin{bmatrix} A & 0 \\ I - F & B_\beta^T \end{bmatrix}$ *has maximum degree η in variable terms $\{a_{x,j}, f_{i,j}\}$, and is linear in each of these variables.*

Proof: Each term $\{a_{x,j}, f_{i,j}, b_{x,j}\}$ appears in only one entry of the Edmonds matrix. Only the η columns corresponding to links carrying random combinations of source processes and/or incoming signals contain variable terms $\{a_{x,j}, f_{i,j}\}$.

The determinant can be written as the sum of products of $r + \nu$ entries, one from each row and column. Each such product is linear in each variable term $\{a_{x,j}, f_{i,j}\}$, and has degree at most η in these variables. ■

Lemma 2 *Let P be a polynomial in $\mathbb{F}[\xi_1, \xi_2, \dots]$ of degree less than or equal to $d\eta$, in which the largest exponent of any variable ξ_i is at most d. Values for $\xi_1, \xi_2, \dots$ are chosen independently and uniformly at random from $\mathbb{F}_q \subseteq \mathbb{F}$. The probability that P equals zero is at most $1 - (1 - d/q)^\eta$ for $d < q$.*

Proof: For any variable ξ_1 in P, let d_1 be the largest exponent of ξ_1 in P. Express P in the form $P = \xi_1^{d_1} P_1 + R_1$, where P_1 is a polynomial of degree at most $d\eta - d_1$ that does not contain variable ξ_1, and R_1 is a polynomial in which the largest exponent of ξ_1 is less than d_1. By the Principle of Deferred Decisions, the probability

$\Pr[P=0]$ is unaffected if we set the value of ξ_1 last after all the other coefficients have been set. If, for some choice of the other coefficients, $P_1 \neq 0$, then P becomes a polynomial in $\mathbb{F}[\xi_1]$ of degree d_1. By the Schwartz-Zippel Theorem, this probability $\Pr[P=0|P_1 \neq 0]$ is upper bounded by d_1/q. So

$$\begin{aligned} \Pr[P=0] &\leq \Pr[P_1 \neq 0]\frac{d_1}{q} + \Pr[P_1 = 0] \\ &= \Pr[P_1 = 0]\left(1 - \frac{d_1}{q}\right) + \frac{d_1}{q}. \end{aligned} \tag{4.1}$$

Next we consider $\Pr[P_1 = 0]$, choosing any variable ξ_2 in P_1 and letting d_2 be the largest exponent of ξ_2 in P_1. We express P_1 in the form $P_1 = \xi_2^{d_2}P_2 + R_2$, where P_2 is a polynomial of degree at most $d\eta - d_1 - d_2$ that does not contain variables ξ_1 or ξ_2, and R_2 is a polynomial in which the largest exponent of ξ_2 is less than d_2. Proceeding similarly, we assign variables ξ_i and define d_i and P_i for $i = 3, 4, \ldots$ until we reach $i = k$ where P_k is a constant and $\Pr[P_k = 0] = 0$. Note that $1 \leq d_i \leq d < q \, \forall \, i$ and $\sum_{i=1}^{k} d_i \leq d\eta$, so $k \leq d\eta$. Applying Schwartz-Zippel as before, we have for $k' = 1, 2, \ldots, k$

$$Pr[P_{k'} = 0] \leq Pr[P_{k'+1} = 0]\left(1 - \frac{d_{k'+1}}{q}\right) + \frac{d_{k'+1}}{q}. \tag{4.2}$$

Combining all the inequalities recursively, we can show by induction that

$$\Pr[P=0] \leq \frac{\sum_{i=1}^{k} d_i}{q} - \frac{\sum_{i\neq j} d_i d_j}{q^2} + \ldots + (-1)^{k-1}\frac{\prod_{i=1}^{k} d_i}{q^k}.$$

Now consider the integer optimization problem

$$\begin{aligned} \text{Maximize} \quad & f = \frac{\sum_{i=1}^{d\eta} d_i}{q} - \frac{\sum_{i\neq j} d_i d_j}{q^2} + \ldots + (-1)^{d\eta-1}\frac{\prod_{i=1}^{d\eta} d_i}{q^{d\eta}} \\ \text{subject to} \quad & 0 \leq d_i \leq d < q \, \forall \, i \in [1, d\eta], \\ & \sum_{i=1}^{d\eta} d_i \leq d\eta, \quad \text{and} \quad d_i \text{ integer} \end{aligned} \tag{4.3}$$

whose maximum is an upper bound on $\Pr[P=0]$.

We first consider the problem obtained by relaxing the integer condition on the variables d_i. Let $\underline{d}^* = \{d_1^*, \ldots, d_{d\eta}^*\}$ be an optimal solution.

For any set S_h of h distinct integers from $[1, d\eta]$, let $f_{S_h} = 1 - \frac{\sum_{i \in S_h} d_i}{q} + \frac{\sum_{i,j \in S_h, i \neq j} d_i d_j}{q^2} - \ldots + (-1)^h \frac{\prod_{i \in S_h} d_i}{q^h}$. We can show by induction on h that $0 < f_{S_h} < 1$ for any set S_h of h distinct integers in $[1, d\eta]$.

If $\sum_{i=1}^{d\eta} d_i^* < d\eta$, then there is some $d_i^* < d$, and there exists a feasible solution $\underline{d}$ such that $d_i = d_i^* + \epsilon$, $\epsilon > 0$, and $d_h = d_h^*$ for $h \neq i$, which satisfies

$$f(\underline{d}) - f(\underline{d}^*) \quad = \quad \frac{\epsilon}{q}\left(1 - \frac{\sum_{h \neq i} d_h^*}{q} + \ldots + (-1)^{d\eta - 1} \frac{\prod_{h \neq i} d_h^*}{q^{d\eta - 1}}\right).$$

This is positive, contradicting the optimality of $\underline{d}^*$.

Next suppose $0 < d_i^* < d$ for some d_i^*. Then there exists some d_j^* such that $0 < d_j^* < d$, since if $d_j^* = 0$ or d for all other j, then $\sum_{i=1}^{d\eta} d_i^* \neq d\eta$. Assume without loss of generality that $0 < d_i^* \leq d_j^* < d$. Then there exists a feasible vector $\underline{d}$ such that $d_i = d_i^* - \epsilon$, $d_j = d_j^* + \epsilon$, $\epsilon > 0$, and $d_h = d_h^*\ \forall\ h \neq i, j$, which satisfies

$$f(\underline{d}) - f(\underline{d}^*) = -\left(\frac{(d_i^* - d_j^*)\epsilon - \epsilon^2}{q^2}\right)\left(1 - \frac{\sum_{h \neq i,j} d_h^*}{q} - \ldots + (-1)^{d\eta - 2} \frac{\prod_{h \neq i,j} d_h^*}{q^{d\eta - 2}}\right).$$

This is again positive, contradicting the optimality of $\underline{d}^*$.

Thus, $\sum_{i=1}^{d\eta} d_i^* = d\eta$, and $d_i^* = 0$ or d. So exactly η of the variables d_i^* are equal to d. Since the optimal solution is an integer solution, it is also optimal for the integer program (4.3). The corresponding optimal $f = \eta \frac{d}{q} - \binom{\eta}{2} \frac{d^2}{q^2} + \ldots + (-1)^{\eta - 1} \frac{d^\eta}{q^\eta} = 1 - \left(1 - \frac{d}{q}\right)^\eta$. ∎

Proof of Theorem 4 for delay-free networks: To check if a network code (A, F, B) transmits all source processes to receiver β, it suffices to check that the determinant of the corresponding Edmonds matrix is nonzero (Theorem 1). This determinant, which we denote by P_β, is a polynomial linear in each variable $\{a_{x,j}, f_{i,j}\}$, with total degree at most η in these variables (Lemma 1). The product $\prod_\beta P_\beta$ for d receivers is, accordingly, a polynomial in $\{a_{x,j}, f_{i,j}\}$ of total degree at most $d\eta$, and in which the largest exponent of each of these variables is at most d.

Recall from the discussion of our model in Chapter 2 that linearly correlated sources can be viewed as pre-specified linear combinations of underlying independent unit entropy rate processes. Unlike the independent sources case where each nonzero entry of the A matrix can be set independently, in this case there are linear dependencies among the entries. The columns of the A matrix are linear functions $\sum_k \alpha_j^k \underline{x}_j^k$ of column vectors $\underline{x}_j^k$ that represent the composition of the source processes at tail(j) in terms of underlying independent processes. In distributed randomized coding, the variables α_j^k are chosen independently and uniformly at random over $\mathbb{F}_q$.

It can seen from Lemma 1 that for any particular j, each product term in the polynomial P_β for any receiver β contains at most one variable $a_{i,j} = \sum_k \alpha_j^k v_{i,j}^k$. P_β is thus linear in the variables α_j^k, and also in variables $f_{i,j}$, which are unaffected by the source correlations. So any variable in the product of d such polynomials has maximum exponent d.

Applying Lemma 2 gives us the required bound.

For the single-receiver case, the bound is attained for a network consisting only of links forming a single set of r disjoint source-receiver paths. ■

These results for acyclic delay-free networks can be generalized to arbitrary networks, as we next show.

4.4 General Networks with Delays

In this section we consider general networks which may have cycles and link delays. As noted earlier, acyclic networks with delay may be operated in a burst-oriented/batch-like fashion which renders the analysis similar to that for acyclic delay-free networks. Here we consider the general cyclic case without buffering, where information is continuously injected into the network. The coefficients of the linear combinations of signals on each link then become polynomials in a delay variable, instead of scalars. The number of terms of these polynomials that must be sent, and the memory required at the receivers, depend on the number of links involved in cycles (memory registers) in the network. For less frequently changing networks, one efficient way to communicate the code coefficients to the receivers following a change is to have a phase in which the sources send a canonical basis through the network.

Lemma 3 *The determinant polynomial of the Edmonds matrix* $\begin{bmatrix} A & 0 \\ I - DF & B_\beta^T \end{bmatrix}$ *associated with a network code* (A, F, B) *in a network with delay is a polynomial in delay variable* D*, whose coefficients have maximum degree* η *in variables* $\{a_{x,j}, f_{i,j}\}$*, and are linear in each variable* $\{a_{x,j}, f_{i,j}\}$*.*

Proof: The proof is analogous to that of the corresponding result (Lemma 1) for delay-free graphs. ■

Proof of Theorem 4 for general networks with delay: To check if a network code (A, F, B) transmits all source processes to receiver β, it suffices to check that the determinant of the corresponding Edmonds matrix is nonzero (Theorem 1). This determinant is a polynomial in delay variable D, whose coefficients are linear in each variable $\{a_{x,j}, f_{i,j}\}$ and have degree at most η in these variables (Lemma 3). The rest of the proof is analogous to that of the corresponding result for acyclic delay-free graphs given in the previous section. ■

While these results hold very generally, they perforce do not take advantage of the particular network structure. However, it is intuitive that redundancy or spare resources in the network should improve the performance of randomized network coding. The next section presents tighter, more specialized bounds for acyclic networks.

4.5 Connections with Link Reliability and Tighter Bounds for Acyclic Graphs

In this section we prove Theorem 5 relating network coding performance and network connection feasibility in the case of unreliable links, which is used in the proof of Theorem 6 quantifying the benefit of spare capacity and effect of unreliable links. While our results in previous sections have all extended to networks with cycles with the introduction of link delays, our proof of Theorem 5 assumes an acyclic network, with or without delays. We have not proven or disproven whether these results extend to networks with cycles.

Lemma 4 *Consider any link j. Let $\underline{v}_i \in (\mathbb{F}_q[D])^r$ be the vector of source coefficients associated with the i^{th} input to link j, and let $Y(j) = \sum_i D f_i \underline{v}_i$ be the vector associated with link j. Consider a number of sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ each consisting of d' arbitrary rank-$(r-1)$ matrices in $(\mathbb{F}_q[D])^{r\times(r-1)}$, such that for each matrix in $\mathcal{S}_k$, $1 \leq k \leq n$, link j has among its inputs a signal whose associated vector is not in the column space of the matrix.*

Denote by $E_{\mathcal{S}_k,j}$ the event that adding $Y(j)$ as an additional column to each of the matrices in $\mathcal{S}_k$ gives a full rank matrix. If coefficients f_i are chosen uniformly and independently from $\mathbb{F}_q$, then $Pr(\bigcup_{k=1}^n E_{\mathcal{S}_k,j}) \geq 1 - d'/q$.

Proof: First consider any one of the sets $\mathcal{S}_{k'}$, $1 \leq k' \leq n$. Each entry of $Y(j)$ is a polynomial in $\mathbb{F}_q[D, f_1, f_2, \ldots]$ that is linear in coefficients f_i. The determinant of an $r \times r$ matrix which has $Y(j)$ as one of its columns, and whose $r-1$ other columns are independent of coefficients f_i, is thus linear in coefficients f_i. The product of d' such determinants has maximum degree d' in coefficients f_i.

By the Schwartz-Zippel Theorem, this product is nonzero with probability at least $1 - d'/q$. Thus, we have $\Pr(E_{\mathcal{S}_{k'},j}) \geq 1 - d'/q$, which gives $\Pr(\bigcup_{k=1}^n E_{\mathcal{S}_k,j}) \geq 1 - d'/q$. ∎

Proof of Theorem 5: Each receiver receives all processes successfully if the submatrix of AG corresponding to r of its incident incoming links, or terminal links, has full rank. The connection problem is feasible if and only if each receiver has a set of r link-disjoint paths, one from each source.

Let j be the highest-indexed link in an ancestral ordering, where lower-indexed links feed into higher-indexed links. Consider any given signals on all other links. There are three cases:

Case 1: Regardless of the code coefficients for j, there cannot exist full rank sets of r terminal links for each receiver.

Case 2: Regardless of the code coefficients for j, each receiver has a full rank set of r terminal links.

Case 3: For some choice of code coefficients for link j, each receiver has a full rank set of r terminal links, i.e. link j has among its inputs signals whose associated vectors are not in the column space of the submatrices of AG corresponding to the other terminal links of one or more receivers. Applying Lemma 4, we see that such a choice is made with probability at least $1 - d'/q$, where d' is the number of receivers downstream of link j.

In all three cases, the probability that each receiver has a set of r terminal links with a full rank set of inputs when code coefficients for link j are chosen randomly is greater than or equal to that in the case where link j is deleted with probability $d/q \geq d'/q$.

We next consider the problem where link j is deleted with probability d/q, and random code coefficients are chosen for all other links. From our earlier arguments, the probability that any set of r undeleted paths to each receiver has a full rank set of inputs is less than or equal to the probability of success in the original network coding problem.

We continue in this fashion, at each stage considering a new problem in which we delete with probability d/q the next highest-indexed link as well as each previously considered link. Random code coefficients are chosen for all other links. At each stage, for any choice of surviving links among the set of randomly deleted links, the problem is either infeasible, or there exist one or more sets of random coding links incident to undeleted paths to each receiver which, if full rank, preserve feasibility of the problem. The probability that any set of r undeleted paths to each receiver has a full rank set of inputs is less than or equal to the probability of success in the original network coding problem.

Note that these arguments hold for the case of independent or linearly correlated sources. ■

Proof of Theorem 6: For a given network of non-failed links, we find a lower bound by considering the case of linearly correlated sources, which includes the case of independent sources as a special case, and by analyzing the probability that the connections remain feasible when links fail with probability $1/q$, which by Theorem 5 gives us a lower bound on network coding success probability. The success probability for a network whose links fail (i.e. are permanently deleted from the network) with probability p is thus lower bounded by the probability that the connections remain feasible when links fail with probability $1 - (1 - p)(1 - 1/q)$.

We show by induction on y that a network consisting of $r + y$ disjoint source-receiver paths of length L, any r of which can transmit all processes, has a success probability that is less than or equal to that for any y-redundant network.

Consider a network $\mathcal{G}_1$ consisting of $r + y$ disjoint source-receiver paths of length L, any r of which can transmit all the processes. Let $\mathcal{G}_2$ be any other y-redundant network with source-receiver paths of length at most L.

For $i = 1, 2$, we consider a set $\mathcal{P}_i$ of links in graph $\mathcal{G}_i$ forming r link-disjoint source-receiver paths sufficient to transmit all processes to the receiver. We distinguish two cases:

Case 1: None of the links in $\mathcal{P}_i$ fail. In this case the connections are feasible.

Case 2: There exists some link $j_i \in \mathcal{P}_i$ that fails.

Then we have

$$\begin{aligned}
\Pr(\text{success}) &= \Pr(\text{case } 1) + \Pr(\text{case } 2) \Pr(\text{success}|\text{case } 2) \\
&= 1 - \Pr(\text{case } 2) \left(1 - \Pr(\text{success}|\text{case } 2)\right).
\end{aligned}$$

Since $\mathcal{P}_1$ has at least as many links as $\mathcal{P}_2$, $\Pr(\text{case } 2, i = 1) \geq \Pr(\text{case } 2, i = 2)$. Thus, if we can show that $\Pr(\text{success}|\text{case } 2, i = 1) \leq \Pr(\text{success}|\text{case } 2, i = 2)$, the induction hypothesis $\Pr(\text{success}|i = 1) \leq \Pr(\text{success}|i = 2)$ follows.

For $y = 0$,the hypothesis is true since $\Pr(\text{success}|\text{case } 2) = 0$ for $i = 1, 2$. For $y > 0$, in case 2 we can remove link j_i leaving a $(y - 1)$-redundant graph $\mathcal{G}'_i$. By the induction hypothesis, the probability of success for $\mathcal{G}'_1$ is less than or equal to that for $\mathcal{G}'_2$.

Thus, $\mathcal{G}_1$ gives a lower bound on success probability, which is the probability that all links on at least r of $r + y$ length-L paths do not fail. The result follows from observing that each path does not fail with probability $\left((1-p)(1-\frac{1}{q})\right)^L$. ■

4.6 Benefits over not using network coding

Having obtained bounds on the performance of randomized network coding, we next consider comparisons against approaches without network coding.

Network coding, as a superset of routing, has been shown to offer significant capacity gains for specially constructed networks [59]. Apart from such examples, however, the capacity gains of centralized network coding over centralized optimal routing have not been as clear.

In this section, we illustrate two types of network scenarios in which distributed randomized coding offers other useful advantages besides capacity gains. The first is in distributed and varying environments where coordination and state maintenance may be expensive or infeasible. An example for which theoretical analysis is tractable is a simple grid topology in the extreme case of no coordination or routing state. One way to operate in this kind of environment is flooding. We give a theoretical comparison of distributed flooding with and without coding for this topology. The second is in networks with dynamically varying connections where online operation is desired. In general, an exact theoretical analysis of optimal multicast routing is difficult, as it is closely related to the NP-complete Steiner-tree problem. Thus, we use simulations to compare network coding with an approximate Steiner-tree heuristic on randomly generated networks.

4.6.1 Distributed Settings

In networks with large numbers of nodes and/or changing topologies, it may be expensive or infeasible to reliably maintain routing state at network nodes. Distributed randomized routing schemes have been proposed [3, 60] which address this kind of issue. However, not allowing different signals to be combined can impose intrinsic penalties in efficiency compared to using network coding.

Consider for example the problem of sending two processes from a source node to receiver nodes in random unknown locations on a rectangular grid network. Transmission to a particular receiver is successful if the receiver gets two different processes instead of duplicates of the same process. Suppose we wish to use a distributed transmission scheme that does not involve any coordination among nodes. Then the best each node can do is to locally try to preserve message diversity, for instance using the following scheme RF (ref Figure 4-2):

- The source node sends one process in both directions on one axis and the other process in both directions along the other axis.
- A node receiving information on one link sends the same information on its three other links (these are nodes along the grid axes passing through the source node).
- A node receiving signals on two links sends one of the incoming signals on one of its two other links with equal probability, and the other signal on the remaining link.

For comparison, we consider the same rectangular grid problem with the following simple random coding scheme RC (ref Figure 4-2):

- The source node sends one process in both directions on one axis and the other process in both directions along the other axis.
- A node receiving information on one link sends the same information on its three other links.
- A node receiving signals on two links sends a random linear combination of the source signals on each of its two other links.[3]

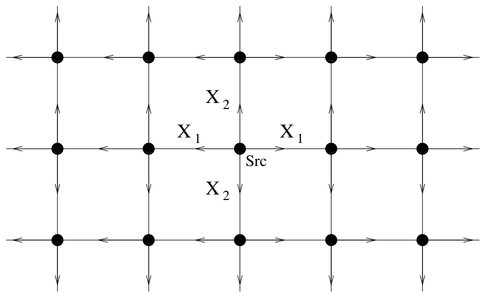


Figure 4-2: Rectangular grid network with two processes X_1 and X_2 originating at a source node.

Receiver position		(2,2)	(3,3)	(4,4)	(10,10)	(2,3)	(9,10)	(2,4)	(8,10)
RF	actual	0.75	0.672	0.637	-	0.562	-	0.359	-
	upper bound	0.75	0.688	0.672	0.667	0.625	0.667	0.563	0.667
RC	$\mathbb{F}_{2^4}$ lower bound	0.772	0.597	0.461	0.098	0.679	0.111	0.597	0.126
	$\mathbb{F}_{2^6}$ lower bound	0.939	0.881	0.827	0.567	0.910	0.585	0.882	0.604
	$\mathbb{F}_{2^8}$ lower bound	0.984	0.969	0.954	0.868	0.977	0.875	0.969	0.882

Table 4.1: Success probabilities of randomized flooding scheme RF and randomized coding scheme RC. The table gives bounds as well as some actual probability values where exact calculations are tractable.

Theorem 7 *For the random flooding scheme RF, the probability that a receiver located at grid position (x, y) relative to the source receives both source processes is at most*

$$\frac{1 + 2^{||x|-|y||+1}(4^{\min(|x|,|y|)-1} - 1)/3}{2^{|x|+|y|-2}}$$

Proof: To simplify notation, we assume without loss of generality that the axes are chosen such that the source is at $(0, 0)$, and $0 < x \le y$. Let $E_{x,y}$ be the event that two different signals are received by a node at grid position (x, y) relative to the source. The statement of the lemma is then

$$\Pr[E_{x,y}] \le \left(1 + 2^{y-x+1}(4^{x-1} - 1)/3\right) / 2^{y+x-2} \tag{4.4}$$

which we prove by induction.

[3]This simple scheme, unlike the randomized flooding scheme RF, leaves out the optimization that each node receiving two linearly independent signals should always send out two linearly independent signals.

Let $Y^h_{x,y}$ denote the signal carried on the link between $(x-1, y)$ and (x, y) and let $Y^v_{x,y}$ denote the signal carried on the link between $(x, y-1)$ and (x, y) (ref Figure 4-3).

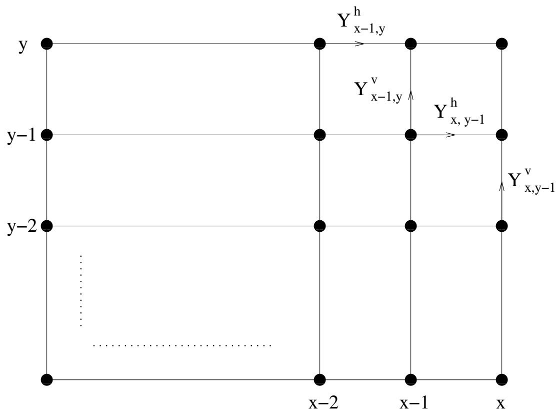


Figure 4-3: Rectangular grid network. $Y^h_{x,y}$ denotes the signal carried on the link between $(x-1, y)$ and (x, y), and $Y^v_{x,y}$ denotes the signal carried on the link between $(x, y-1)$ and (x, y).

Observe that $\Pr[E_{x,y}|E_{x-1,y}] = 1/2$, since with probability 1/2 node $(x-1, y)$ transmits to node (x, y) the signal complementary to whatever signal is being transmitted from node $(x, y-1)$. Similarly, $\Pr[E_{x,y}|E_{x,y-1}] = 1/2$, so $\Pr[E_{x,y}|E_{x-1,y}$ or $E_{x,y-1}] = 1/2$.

Case 1: $E_{x-1,y-1}$

Case 1a: $Y^h_{x-1,y} \neq Y^v_{x,y-1}$. With probability $\frac{1}{2}$, $Y^v_{x-1,y} \neq Y^h_{x-1,y}$, resulting in $E_{x,y-1} \cup E_{x-1,y}$. With probability $\frac{1}{2}$, $Y^v_{x,y-1} = Y^h_{x,y-1}$, resulting in $E_{x,y}$. So $\Pr[E_{x,y}|$ Case 1a$] = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \frac{3}{4}$.

Case 1b: $Y^h_{x-1,y} = Y^v_{x,y-1}$. Either $E_{x,y-1} \cup \overline{E}_{x-1,y}$ or $\overline{E}_{x,y-1} \cup E_{x-1,y}$, so $\Pr[E_{x,y}|$ Case 1b$] = 1/2$.

Case 2: $\overline{E}_{x-1,y-1}$

Case 2a: $Y^h_{x-1,y} \neq Y^v_{x,y-1}$. Either $E_{x,y-1} \cup \overline{E}_{x-1,y}$ or $\overline{E}_{x,y-1} \cup E_{x-1,y}$, so $\Pr[E_{x,y}|$ Case 2a$] = 1/2$.

Case 2b: $Y^h_{x-1,y} = Y^v_{x,y-1} = Y^h_{x-1,y-1}$. By the assumption of case 2, $Y^v_{x,y-1}$ is also equal to this same signal, and $\Pr[E_{x,y}|$ Case 2b$] = 0$.

Case 2c: $Y^h_{x-1,y} = Y^v_{x,y-1} \neq Y^h_{x-1,y-1}$. Then $E_{x,y-1}$ and $E_{x-1,y}$, so $\Pr[E_{x,y}|$ Case 2c$] = 1/2$.

So

$$\begin{aligned}
\Pr[E_{x,y}|E_{x-1,y-1}] &\leq \max\left(\Pr[E_{x,y}|\text{ Case 1a}], \Pr[E_{x,y}|\text{ Case 1b}]\right) \\
&= 3/4 \\
\Pr[E_{x,y}|\overline{E}_{x-1,y-1}] &\leq \max\left(\Pr[E_{x,y}|\text{ Case 2a}],\ \Pr[E_{x,y}|\text{ Case 2b}],\ \Pr[E_{x,y}|\text{ Case 2c}]\right) \\
&= 1/2 \\
\Pr[E_{x,y}] &\leq \frac{3}{4}\Pr[E_{x-1,y-1}] + \frac{1}{2}\Pr[\overline{E}_{x-1,y-1}] \\
&= \frac{1}{2} + \frac{1}{4}\Pr[E_{x-1,y-1}]
\end{aligned}$$

If (4.4) holds for some (x, y), then it also holds for $(x + 1, y + 1)$:

$$\begin{aligned}
\Pr[E_{x+1,y+1}] &\leq \frac{1}{2} + \frac{1}{4}\Pr[E_{x,y}] \\
&= \frac{1}{2} + \frac{1}{4}\left(\frac{1 + 2^{y-x+1}(1 + 4 + \ldots + 4^{x-2})}{2^{y+x-2}}\right) \\
&= \frac{1 + 2^{y-x+1}(4^x - 1)/3}{2^{y+1+x+1-2}}
\end{aligned}$$

Now $\Pr[E_{1,y'}] = 1/2^{y'-1}$, since there are $y'-1$ nodes, $(1, 1), \ldots, (1, y'-1)$, at which one of the signals being transmitted to $(1, y')$ is eliminated with probability $1/2$. Setting $y' = y - x + 1$ gives the base case which completes the induction. ■

Theorem 8 *For the random coding scheme RC, the probability that a receiver located at grid position (x, y) relative to the source can decode both source processes is at least $(1 - 1/q)^{2(x+y-2)}$.*

Proof: We first establish the degree of the polynomial P_β for a receiver β at (x, y), in the indeterminate variables $f_{i,j}$. By Theorem 3, P_β is a linear combination of product terms of the form $a_{1,l_1} a_{2,l_2} f_{i_1,l_3} \ldots f_{i_l,l_k}$, where $\{l_1, \ldots, l_k\}$ is a set of distinct links forming two disjoint paths from the source to the receiver. In the random coding scheme we consider, the only randomized variables are the $f_{i,j}$ variables at nodes receiving information on two links. The maximum number of such nodes on a source-receiver path is $x + y - 2$, so the total degree of P_β is $2(x + y - 2)$. Applying the random coding bound of Lemma 2 yields the result. ■

Table 4.6.1 gives, for various values of x and y, the values of the success probability bounds as well as some actual probabilities for randomized flooding when x and y are small. Note that an increase in grid size from 3×3 to 10×10 requires only an increase of two in codeword length to obtain success probability lower bounds close to 0.9, which are substantially better than the upper bounds for routing.

4.6.2 Dynamically Varying Connections

Another scenario we consider is an online multi-source multicast problem in which sources turn on and off dynamically, comparing distributed randomized coding to an approximate online Steiner tree routing approach from [37] in which, for each transmitter, a tree is selected in a centralized fashion.

Multicast connection requests are presented and accommodated sequentially. Existing connections are not disrupted or rerouted in trying to accommodate new requests. The algorithms are evaluated on the basis of the number of connections that are rejected or blocked owing to capacity limitations, and the multicast throughput supported.

For simplicity, we run our trials on directed acyclic networks, assuming that there exist mechanisms, e.g. based on geographical position or connection information, to avoid transmitting information in cycles. We also assume integer edge capacities and integer source entropy rates.

The online routing algorithm we consider finds a multicast tree for each new source using the Nearest Node First (NNF) heuristic for Steiner tree computation from [37], which uses Dijkstra's shortest path algorithm to reach receiver nodes in order of increasing distance from the source. Dijkstra's shortest path algorithm is run until a receiver node is reached. The corresponding source-receiver path is added to the Steiner tree and the costs of all the edges along this path are set to zero. The algorithm is then applied recursively on the remaining receiver nodes.

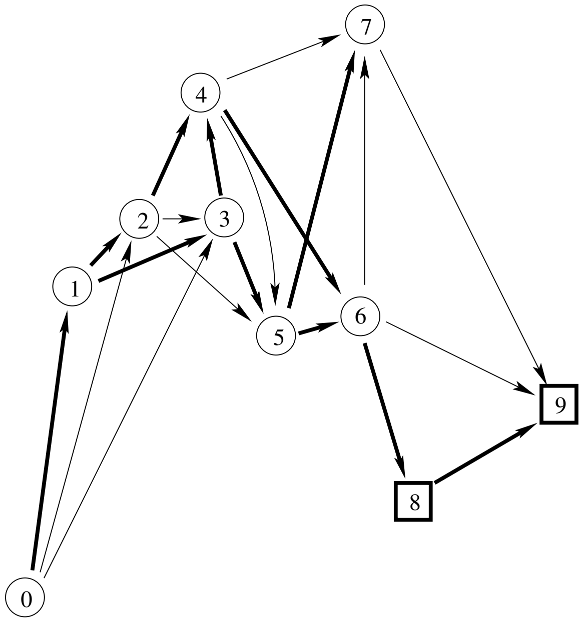


Figure 4-4: An example of a randomly generated network used in our trials. This network was generated with parameters $n = 10$, $s = 6$, $r = 2$, $i = 4$, $\rho = 0.6$. Nodes are labeled as circles, and the receivers are squares; thick lines denote links with capacity two, and thin lines denote links with capacity one.

4.6.3 Experimental setup

We run our trials on randomly generated geometric graphs. Test networks are generated with the following parameters: number of nodes n, number of sources r, number of receivers d, transmission range ρ, maximum in-degree and out-degree i. The parameter values for the tests are chosen such that the resulting random graphs would in general be connected and able to support some of the desired connections, while being small enough for the simulations to run efficiently. For each trial, n nodes are scattered uniformly over a unit square. To create an acyclic graph we order the nodes by their x-coordinate and choose the direction of each link to be from the lower numbered to the higher numbered node. Any pair of nodes within a distance ρ of each other is connected by a unit capacity link, and any pair within distance $\rho/\sqrt{2}$

Table 4.2: A sample of results on graphs generated with the following parameters: number of nodes n, number of sources r, number of receivers d, transmission range ρ, maximum in-degree and out-degree i. b_r and b_c are the rate of blocked connections for routing and coding, respectively, and t_r and t_c are the corresponding throughputs.

Parameters						Results				
nodes n	srcs s	rcvrs d	deg i	range ρ	prob p_O	Network	b_r	t_r	b_c	t_c
8	6	1	4	0.5	0.6	1	1.54	1.46	1.55	1.46
						2	0.72	2.27	0.74	2.31
						3	0.26	2.78	0.23	2.74
9	6	2	3	0.5	0.7	1	2.14	0.84	2.17	0.83
						2	0.70	2.31	0.68	2.28
						3	0.90	2.05	0.71	2.26
10	4	2	4	0.5	0.6	1	0.61	1.43	0.50	1.45
						2	1.62	0.53	1.52	0.54
						3	0.14	1.96	0.00	2.05
10	6	2	4	0.5	0.5	1	1.31	1.63	0.71	2.28
						2	0.74	2.17	0.64	2.42
						3	1.51	1.54	1.49	1.61
10	9	3	3	0.5	0.7	1	1.05	2.37	1.14	2.42
						2	1.36	2.22	1.06	2.39
						3	2.67	0.87	2.56	0.89
12	6	2	4	0.5	0.6	1	1.44	1.67	0.71	2.31
						2	0.28	2.72	0.29	2.75
						3	0.75	2.28	0.73	2.31
12	8	2	3	0.5	0.7	1	2.39	1.73	2.34	1.74
						2	2.29	1.73	2.23	1.74
						3	1.57	2.48	1.52	2.51

of each other is connected by a link of capacity 2, provided this does not violate the degree constraints. The receiver nodes are chosen to be the d highest numbered nodes, and r source nodes are chosen randomly (with replacement) from among the lower-numbered half of the nodes. An example topology is given in Figure 4-4.

Each trial consists of a number of periods during which each source is either on (i.e. is actively transmitting) or off (i.e. not transmitting). During each period, any currently-on source turns off with probability p_O, and any currently-off source turns on with probability p_O if it is able to reach all the receivers. A source that is unable to reach all the receivers is blocked from turning on.

Initially all sources are off. For routing, in order for a source to turn on, it would need to find a tree connecting it to all the receivers using spare network capacity unreserved by other sources, and would then reserve capacity corresponding to the tree. A source that turns off frees up its reserved links for new connections. For coding, each network node that tries to turn on initiates up to three random choices of code coefficients within the network. If the receivers are able to decode the new source in addition to all the sources that are already on, the new source is allowed to turn on. A source that is not allowed to turn on is considered a blocked request.

The frequency of blocked requests and the average throughput are calculated for windows of 250 periods until these measurements reach steady-state, i.e. measurements in three consecutive periods being within a factor of 0.1 from each other. This avoids transient initial startup behavior.

4.6.4 Results and discussion

We ran simulations on 242 networks generated randomly using 45 different parameter combinations. In 44 of these networks, coding outperformed routing in both blocking rate and throughput, doing better by more than 10% in at least one of these parameters. In 15 of these, coding outperformed routing in both parameters by more than 10%. In the rest, routing and coding showed comparable performance. Some results for various randomly generated networks are given in table 4.2.

These simulations do not attempt to quantify precisely the differences in performance and overhead of randomized coding and online routing. However, they serve as useful illustrations in two ways.

Firstly, they show that the performance of the Steiner tree heuristic is exceeded by randomized coding over a non-negligible proportion of our randomly constructed graphs, indicating that when connections vary dynamically, coding offers advantages that are not circumscribed to carefully constructed examples. This is in contrast to static settings with optimal centralized control.

Secondly, the simulations illustrate the kinds of field sizes needed in practice for networks with a moderate number of nodes. Field size is important, since it affects memory and complexity requirements. To this end, the simulations use a small field size that still allows randomized coding to generally match the performance of the Steiner heuristic, and to surpass it in networks whose topology makes coding desirable over trees. The theoretical bounds of previous sections guarantee the optimality of randomized coding for large enough field sizes, but they are tight for worst-case network scenarios. In our trials, a field size of 17 with up to three retrials proved sufficient to achieve equal or better performance compared to the Steiner heuristic. The simulations show the applicability of short network code lengths for moderately-sized networks.

4.7 Discussion

Our work represents a first exploration on randomized network coding, giving rise to many questions for further research. We do not consider aspects such as resource and energy allocation, but focus on optimally exploiting a given set of resources. There are also many issues we have not addressed surrounding the adaptation of protocols, which generally assume routing, to randomized coding. Our aim here is to demonstrate the potential benefits of randomized network coding, motivating future consideration of protocol compatibility with or adaptation to network codes.

The basic randomized network coding approach requires no coordination among nodes. If we allow for retrials to find successful codes, we in effect trade code length for some rudimentary coordination. Implementations for various applications may not be completely protocol-free, but the roles and requirements for protocols may be substantially redefined in this new environment.

Chapter 5

Byzantine modification detection with distributed randomized network coding

Overlay multicast or ad hoc multicast represent natural areas of application for the distributed randomized network coding approach of the previous chapter. In overlay or ad hoc multicast settings, end hosts help to forward packets to other end hosts. Such networks are thus more susceptible to Byzantine (i.e., arbitrary) attacks from compromised end hosts, which have access to the same information as other end hosts, and can forward to them arbitrarily modified information.

In this chapter, we show that Byzantine modification detection capability can be added to a multicast scheme based on randomized network coding, with minimal additional computational and communication overhead, by incorporating a simple polynomial hash value in each packet. We consider multi-source multicast mesh networks with multiple paths between the sources and each receiver. The key insight in our approach is that path diversity, coupled with the randomized and distributed choice of codes, makes it hard for an attacker to observe or predict the exact combinations of source information in all other packets received at the receivers. With our approach, a receiver can detect Byzantine modifications with high probability, as long as these modifications have not been designed using knowledge of all the other packets obtained at the receiver.

The detection probability can be traded off against the overhead (i.e., the ratio of hash bits to data bits) – the detection probability increases with the overhead, as well as with the number of unmodified packets obtained at the receiver whose contents are unknown to the attacker. Depending on the application, various responses may be employed upon detection of a Byzantine fault, such as collecting more packets from different nodes to obtain a consistent decoding set, or employing a more complex Byzantine agreement algorithm to identify the Byzantine node(s).

We are able to use a simple polynomial function instead of a complex cryptographic hash function (such as MD5) because our scheme's effectiveness depends only on the fact that there are independent sources of randomness, not all of which are known to the attacker because of path diversity. The use of a simple polynomial function is desirable because it incurs less computational overhead than existing cryptographic hashes.

5.1 Background and related work

The Byzantine problem was first formalized in [41], and has been studied extensively in a variety of contexts such as reliable distributed networks of processors [19, 7] and secure network communications [46, 36, 54, 53, 11]. These and other existing works generally either use cryptographic functions, multiple rounds of message passing or some combination of the two to detect and recover from Byzantine faults. References [7] and [36] optimize for normal performance by using less complex message authentication codes and signed digests respectively during normal operation, resorting to more complex recovery mechanisms only upon detection of a fault. Our technique allows for detection without the use of any cryptographic functions (thereby incurring little computation overhead), and can similarly be used in conjunction with more complex recovery techniques which are activated upon detection of a Byzantine fault.

The reliance on random values unknown to the attacker is reminiscent of one-time pads [47], but our scheme is different because the one-time pad provides secrecy and not authenticity[1], while our scheme aims to provide the latter. Also, unlike one-time pads, the burden of generating the random values is distributed over the network rather than falling solely on the source. Cai and Yeung [6] have studied the problem of providing secrecy in a network coding setting.

5.2 Model

Consider a set of r source packets which are coded together and multicast, using distributed randomized network coding in the finite field $\mathbb{F}_q$. Let the data content of each packet be represented by b elements from $\mathbb{F}_q$, and the hash value by c elements from the same field, and let row vector $\underline{m_i} \in \mathbb{F}_q^{(b+c)}$ represent the concatenation of the data and corresponding hash value for packet i. We denote by M the matrix whose i^{th} row is $\underline{m_i}$.

A genuine, or unmodified, packet contains a random linear combination of one or more of these vectors, along with the coefficients of the combination. This information, for a set $\mathcal{U}$ of unmodified packets, can be represented as the matrix product $C(\mathcal{U})\,[M|I]$, where the coefficient matrix $C(\mathcal{U})$ for the set $\mathcal{U}$ is defined as the $|\mathcal{U}| \times r$ matrix whose i^{th} row is the vector of code coefficients of the i^{th} packet. Decoding for a set $\mathcal{U}$ of r linearly independent packets corresponds to pre-multiplying the associated matrix $C(\mathcal{U})\,[M|I]$ with $C(\mathcal{U})^{-1}$.

Modified packets may contain arbitrary data and hash values. A set of modified packets can be represented in general by $[C_mM + V|C_m]$, where V is an arbitrary $(r-s) \times (b+c)$ matrix. Inconsistent data and hash values, i.e. $V \neq 0$, will cause the decoded packets to differ from the original packets.

Suppose the receiver tries to decode using s unmodified packets and $r-s$ modified packets, where $1 \leq s \leq r-1$. Let C_u and C_m be the coefficient matrices of the set of unmodified packets and the set of modified packets respectively, and let $C = \left[\begin{array}{c} C_u \\ \hline C_m \end{array}\right]$.

[1] Secrecy and authenticity are known to be independent attributes of a cryptographic system [47].

The receiver's decoding process is equivalent to pre-multiplying the matrix

$$\left[\begin{array}{c|c} C_u M & C_u \\ \hline C_m M + V & C_m \end{array}\right] = \left[\begin{array}{c|c} CM + \left[\begin{array}{c} 0 \\ \hline V \end{array}\right] & C \end{array}\right]$$

with C^{-1}. This gives

$$\left[\begin{array}{c|c} M + C^{-1}\left[\begin{array}{c} 0 \\ \hline V \end{array}\right] & I \end{array}\right]$$

i.e., the receiver decodes to $M + \Delta M$, where

$$\Delta M = C^{-1}\left[\begin{array}{c} 0 \\ \hline V \end{array}\right] \tag{5.1}$$

gives the disparity between the decoded packets and the original packets.

5.3 Main results

Consider a Byzantine attacker that supplies modified packets, without knowing the contents of $s \geq 1$ genuine unmodified packets that will be part of a set of r packets used for decoding at a receiver. This is a reasonable assumption given the distributed randomness and path diversity of the network coding setup we consider. The only essential condition is that the attacker does not create its packets knowing the contents of all other packets used for decoding, which makes our results very general: they apply regardless of whether the attacker knows which or how many of its own packets will be used for decoding, and whether there are some unmodified packets whose contents are known to the attacker.

Let ω be the rank of the matrix V, defined in the previous section, that represents the modifications.

The following result characterizes the family of potential outcomes of decoding from the set of packets– the attacker cannot narrow down the set of possible outcomes beyond this regardless of how it designs its modified packets.

Theorem 9 *The attacker cannot determine which of a set of $q^{s\omega}$ potential decoding outcomes the receiver will obtain. In particular, there will be at least s packets such that, for each of these, the attacker knows only that the vector representation of its decoded value will be one of q^{ω} possibilities $\{\underline{m}_i + \sum_{j=1}^{\omega} \gamma_{i,j}\underline{v}_j | \gamma_{i,j} \in \mathbb{F}_q\}$, where $\underline{m}_i$ is the vector representation of the data and hash value of some original packet, and $\underline{v}_j$ is determined by the attacker's modifications.* □

The next result provides, for a simple polynomial hash function, an upper bound on the proportion of potential decoding outcomes that can have consistent data and hash values, in terms of $\kappa = \lceil \frac{b}{c} \rceil$, the ceiling of the ratio of the number of data symbols to hash symbols. Larger values for k correspond to lower overheads but higher probability of a successful attack. This tradeoff is a design parameter for the network.

Theorem 10 *Suppose each packet contains b data symbols $x_1, \ldots, x_b$ and $c \le b$ hash symbols $y_1, \ldots, y_c$. Consider the function $h : \mathbb{F}_q^k \rightarrow \mathbb{F}_q$ mapping $(x_1, \ldots, x_k)$, $x_i \in \mathbb{F}_q$, to $h(x_1, \ldots, x_k) = x_1^2 + \ldots + x_k^{k+1}$ for any positive integer k. If y_i is set to $h(x_{(i-1)\kappa+1}, \ldots, x_{i\kappa})$ for $i = 1, \ldots, c-1$ and y_c to $h(x_{(c-1)\kappa+1}, \ldots, x_b)$, then the decoded packets can have consistent data and hash values under at most a fraction $\left(\frac{\kappa+1}{q}\right)^s$ of potential values of the unmodified packets, or, from an alternate viewpoint, at most a fraction $\left(\frac{\kappa+1}{q}\right)^s$ of potential outcomes can have consistent data and hash values.* □

Corollary 1 *If the receiver obtains more than r packets, it can use all the packets by decoding from more than one set. If $s' \ge 1$ of the packets are unmodified and have contents that are unknown to the attacker, then the decoded packets can have consistent data and hash values under at most a fraction $\left(\frac{\kappa+1}{q}\right)^{s'}$ of potential values of the unmodified packets (at most a fraction $\left(\frac{\kappa+1}{q}\right)^{s'}$ of potential outcomes can have consistent data and hash values).*

5.4 Detailed development, proofs and ancillary results

5.4.1 Vulnerable scenario

Before proving the results stated in the previous section, we first point out that this approach does not apply in the case where the attacker knows, or has information allowing it to predict with reasonable probability, that it is the only node supplying information to a receiver on a particular subset of the original packets. In such a case, this kind of non-cryptographic scheme cannot prevent the attacker from supplying spurious packets with consistent data and hash values. However, such a scenario is unlikely to persist if sources are reasonably well connected, and nodes periodically and randomly switch their connections among neighboring nodes.

Mathematically, this case corresponds to the attacker knowing that a particular set of columns of any potential matrix C_u for the receiver will be zero. Without loss of generality, assume that the last $t \leq r - s$ columns of C_u are zero. The attacker can then make C a block diagonal matrix by choosing C_m to be of the form $\left[\begin{array}{c|c} C'_m & 0 \\ \hline 0 & C''_m \end{array}\right]$, where C''_m is a $t \times t$ matrix and the rows of $\left[\begin{array}{c|c} C'_m & 0 \end{array}\right]$ are independent of the rows of C_u. Then C^{-1} is also block diagonal, of the form

$$\left[\begin{array}{c|c} \left[\begin{array}{c} C'_u \\ \hline C'_m \end{array}\right]^{-1} & 0 \\ \hline 0 & C''^{\,-1}_m \end{array}\right].$$

Since C''_m is determined by the attacker, it can choose ΔM by setting $V = \left[\begin{array}{c} 0 \\ \hline V' \end{array}\right]$, where V' is an appropriately chosen $t \times (b + c)$ matrix.

5.4.2 Protected scenario

We next consider the case where the attacker does not know the contents of other packets the receiver will use for decoding. In this case, it designs its packets, i.e. fixes C_m and V, knowing only that $\begin{bmatrix} C_u \\ \hline C_m \end{bmatrix}$ is nonsingular.

Proof of Theorem 9: Consider any fixed C_m and V. A receiver decodes only when it has a set of packets such that corresponding coefficients of the matrix C is non-singular. Therefore, we consider the set $\mathcal{Z}$ consisting of the values of C_u that satisfy the condition that $C = \begin{bmatrix} C_u \\ \hline C_m \end{bmatrix}$ is nonsingular.

We show that we can partition the set $\mathcal{Z}$ into cosets

$$\mathcal{Z}_i = \{C_i + RC_m | R \in \mathbb{F}_q^{s\times(r-s)}\}, i = 1, 2, \ldots, \chi$$

where

$$\begin{aligned} \chi &= \frac{|\mathcal{Z}|}{q^{s(r-s)}} \\ &= \frac{\prod_{k=0}^{s-1}(q^r - q^{r-s+k})}{q^{s(r-s)}} \\ &= q^{s(s-1)/2}\prod_{k=1}^{s}(q^k - 1). \end{aligned}$$

Then we show that each coset can be further partitioned into equal-sized sets that each generate, via (5.1), the full set of possible modifications ΔM. Hence, it suffices to focus on just one of these subsets of $\mathcal{Z}$ in proving Theorem 9.

To see that we can partition $\mathcal{Z}$ into cosets as asserted above, consider the following procedure for constructing such cosets: Any element of $\mathcal{Z}$ can be chosen as C_1, giving coset $\mathcal{Z}_1 = \{C_i + RC_m | R \in \mathbb{F}_q^{s\times(r-s)}\}$. Next, $C_2, C_3, \ldots, C_\chi$ are chosen sequentially to be any element of $\mathcal{Z}$ not in the cosets $\mathcal{Z}_j$ of previously chosen elements. Note that this forms a partition of $\mathcal{Z}$, since the presence of some element c in two sets $\mathcal{Z}_i$ and $\mathcal{Z}_j$ implies that C_j is also in $\mathcal{Z}_i$, which is a contradiction. It is also clear that each coset has size

$$\left|\{R|R \in \mathbb{F}_q^{s\times(r-s)}\}\right| = q^{s(r-s)}$$

since C_m has full row rank.

For each such coset $\mathcal{Z}_{\hat{i}}$, the corresponding values of ΔM satisfy, from (5.1),

$$\begin{aligned}
\left(\begin{bmatrix} C_{\hat{i}} \\ \hline C_m \end{bmatrix} + \begin{bmatrix} RC_m \\ \hline 0 \end{bmatrix}\right)\Delta M &= \begin{bmatrix} 0 \\ \hline V \end{bmatrix} \\
\begin{bmatrix} C_{\hat{i}} \\ \hline C_m \end{bmatrix}\Delta M &= \begin{bmatrix} -R \\ \hline I \end{bmatrix} V \\
\Delta M &= \begin{bmatrix} C_{\hat{i}} \\ \hline C_m \end{bmatrix}^{-1}\begin{bmatrix} -R \\ \hline I \end{bmatrix} V
\end{aligned}$$

where each entry $r_{i,j} = R(i,j)$ of $R \in \mathbb{F}_q^{s\times(r-s)}$ is, to the attacker, an unknown variable that can take potentially any value in $\mathbb{F}_q$.

We note that even within one of these cosets $\mathcal{Z}_{\hat{i}}$, multiple values of R will map to the same value of ΔM if V has dependent rows. If so, we further partition each coset into subsets such that the elements in each subset are in one-to-one correspondence with the full set of possible values for ΔM, as we describe below.

Consider a set of ω independent rows of V (where ω is the rank of V, defined previously). Denote by $\mathcal{I}$ the corresponding set of row indexes, and denote by $V_{\mathcal{I}}$ the submatrix consisting of those rows. Each row $\underline{v}_j$ of V can be represented as a linear combination $\underline{v}_j = \sum_{k=1}^{\omega} l_{j,k}\underline{v}_k$ of rows of $V_{\mathcal{I}}$. The coefficients $l_{j,k}$ can be collected into an $(r-s)\times\omega$ matrix L whose $(j,k)^{th}$ entry is $l_{j,k}$, which satisfies

$$V = LV_{\mathcal{I}}$$

We define $R_{\mathcal{I}} = RL$, noting that

$$R_{\mathcal{I}}V_{\mathcal{I}} = RLV_{\mathcal{I}} = RV$$

Note also that the submatrix of L consisting of its ω rows corresponding to set $\mathcal{I}$ is an identity matrix. Thus, each variable $r_{i,j}, j \in \mathcal{I}$, appears in exactly one entry of $R_{\mathcal{I}}$ as part of a linear combination with one or more variables $r_{i,j}, j \notin \mathcal{I}$. It follows that $R_{\mathcal{I}}$ can take potentially any value in $\mathbb{F}_q^{s\times\omega}$, and every row of $R_{\mathcal{I}}V_{\mathcal{I}}$ can take on any value in the row space of V. Furthermore, the possible values of R can be partitioned into equal-sized sets, each of which contains all values $\tilde{R} \in \mathbb{F}_q^{s\times(r-s)}$ such that $\tilde{R}L$ equals some particular value $\tilde{R_{\mathcal{I}}}$. The $q^{s\omega}$ possible values for $R_{\mathcal{I}}$ give rise to $q^{s\omega}$ distinct values for $R_{\mathcal{I}}V$, which give in turn $q^{s\omega}$ distinct values for ΔM.

We note that the set of values

$$\left\{ \left[\begin{array}{c} C_{\hat{i}} \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R \\ \hline I \end{array}\right] V \,\middle|\, R \in \mathbb{F}_q^{s\times(r-s)} \right\}$$
$$= \left\{ \left[\begin{array}{c} C_{\hat{i}} \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right] V_{\mathcal{I}} \,\middle|\, R_{\mathcal{I}} \in \mathbb{F}_q^{s\times\omega} \right\}$$

corresponding to any single coset $\mathcal{Z}_{\hat{i}}$ is in one-to-one correspondence with that of any other coset. To see this, observe that for any fixed $R_{\mathcal{I}} \in \mathbb{F}_q^{s\times\omega}$ and fixed distinct C_u, C'_u, we obtain the same values for $\Delta M = \left[\begin{array}{c} C_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right] V_{\mathcal{I}}$ and $\Delta M' = \left[\begin{array}{c} C'_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R'_{\mathcal{I}} \\ \hline L \end{array}\right] V_{\mathcal{I}}$ by setting

$$R'_{\mathcal{I}} = -C'_u \left[\begin{array}{c} C_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right]$$

which gives:

$$\Delta M' \;=\; \left[\begin{array}{c} C'_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} C'_u \left[\begin{array}{c} C_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right] V_{\mathcal{I}} \\ \hline LV_{\mathcal{I}} \end{array}\right]$$

$$
\begin{aligned}
&= \left[\begin{array}{c} C'_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} C'_u \Delta M \\ \hline V \end{array}\right] \\
&= \left[\begin{array}{c} C'_u \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} C'_u \\ \hline C_m \end{array}\right] \Delta M \\
&= \Delta M
\end{aligned}
$$

These observations allow us to focus on a single set

$$
\left\{ \left[\begin{array}{c} C_{\hat{i}} \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right] V_{\mathcal{I}} \;\middle|\; R_{\mathcal{I}} \in \mathbb{F}_q^{s \times \omega} \right\}
$$

corresponding to any coset $\mathcal{Z}_{\hat{i}}$.

Let $\left[\begin{array}{c} C_{\hat{i}} \\ \hline C_m \end{array}\right]^{-1} \left[\begin{array}{c} -R_{\mathcal{I}} \\ \hline L \end{array}\right]$ be denoted by S. Each row of S is an linear function of one or more rows of $R_{\mathcal{I}}$, either constant, or else dependent on $R_{\mathcal{I}}$ and taking potentially any value in $\mathbb{F}_q^{\omega}$. Since $\left[\begin{array}{c} C_{\hat{i}} \\ \hline C_m \end{array}\right]^{-1}$ is nonsingular, at least s rows of S are dependent on $R_{\mathcal{I}}$. The corresponding rows of $SV_{\mathcal{I}}$ are also dependent on $R_{\mathcal{I}}$; for the i^{th} of these rows, the potential values form a set $\{\sum_{j=1}^{\omega} \gamma_{i,j} \underline{v}_j | \gamma_{i,j} \in \mathbb{F}_q\}$, where vector $\underline{v}_j$ corresponds to the j^{th} row of $V_{\mathcal{I}}$. The potential values of the corresponding decoded packets then form a set $\{\underline{m}_i + \sum_{j=1}^{\omega} \gamma_{i,j} \underline{v}_j | \gamma_{i,j} \in \mathbb{F}_q\}$, where $\underline{m}_i$ is the vector representation of the data and hash value of the i^{th} packet. ■

The following lemma is useful in the proof of Theorem 10.

Lemma 5 *Consider the following hash function $h : \mathbb{F}_q^k \to \mathbb{F}_q$ mapping $(x_1, \ldots, x_k)$, $x_i \in \mathbb{F}_q$, to $h(x_1, \ldots, x_k) = x_1^2 + \ldots + x_k^{k+1}$, and denote by $\mathcal{S}(\underline{u}, \underline{v})$ the set of vectors $\{\underline{u} + \gamma\underline{v} | \gamma \in \mathbb{F}_q\}$, where $\underline{u}$ and $\underline{v}$ are fixed vectors. At most $k+1$ out of the q vectors in a set $\mathcal{S}(\underline{u}, \underline{v})$, where $\underline{u} = (u_1, \ldots, u_{k+1})$ is a fixed length-$(k+1)$ vector and $\underline{v} = (v_1, \ldots, v_{k+1})$ a fixed nonzero length-$(k+1)$ vector, can satisfy the property that the last element of the vector equals the hash of the first k elements.*

Proof: Suppose some vector $\underline{u} + \gamma\underline{v}$ satisfies this property, i.e.

$$u_{k+1} + \gamma v_{k+1} = (u_1 + \gamma v_1)^2 + \ldots + (u_k + \gamma v_k)^{k+1} \tag{5.2}$$

Note that for any fixed value of $\underline{u}$ and any fixed nonzero value of $\underline{v}$, (5.2) is a polynomial equation in γ of degree equal to $1 + \tilde{k}$, where $\tilde{k} \in [1, k]$ is the highest index for which the corresponding $v_{k'}$ is nonzero, i.e. $v_{\tilde{k}} \neq 0, v_{k'} = 0 \ \forall\ k' > \tilde{k}$. By the fundamental theorem of algebra, this equation can have at most $1 + \tilde{k} \leq 1 + k$ roots. Thus, the property can be satisfied for at most $1 + k$ values of γ. ■

Proof of Theorem 10: Each hash symbol is used to protect $k \leq \kappa$ data symbols. We consider the set of possible outcomes when a modification is made to at least one symbol of a set consisting of k data symbols and their corresponding hash symbol.

Continuing from the proof of the Theorem 9, we note that S contains s rows that are independent linear combinations of rows of $R_{\mathcal{I}}$. For any particular values of a subset of these rows, each of the remaining rows can take potentially any value in $\mathbb{F}_q^{\omega}$. We consider each of the corresponding rows of $SV_{\mathcal{I}}$ in turn, noting that the set of potential values for the i^{th} of these rows, for any particular values of previously considered rows, is of the form $\{\sum_{j=1}^{\omega} \gamma'_{i,j}\underline{v}_j | \gamma'_{i,j} \in \mathbb{F}_q\}$, and that the set of potential values of the corresponding decoded packets is of the form $\{\underline{m}_i + \sum_{j=1}^{\omega} \gamma'_{i,j}\underline{v}_j | \gamma'_{i,j} \in \mathbb{F}_q\}$.

If $\omega > 1$, the q^{ω}-element set $\{\underline{m}_i + \sum_{j=1}^{\omega} \gamma_{i,j}\underline{v}_{i,j} | \gamma_{i,j} \in \mathbb{F}_q\}$ can be partitioned into $q^{\omega-1}$ size-q sets $\{\underline{m}_i + \sum_{j=1}^{\omega-1} \gamma_{i,j}\underline{v}_{i,j} + \gamma_{i,\omega}\underline{v}_{i,\omega} | \gamma_{i,\omega} \in \mathbb{F}_q\}$, where each set corresponds to a different set of values for $\gamma_{i,1}, \ldots, \gamma_{i,\omega-1}$.

Applying Lemma 5 to each set $\mathcal{S}(\underline{m}_i + \sum_{j=1}^{\omega-1} \gamma_{i,j}\underline{v}_{i,j}, \underline{v}_{i,\omega})$ gives the desired result. Note that the case where $\underline{v} = \underline{0}$ corresponds to the trivial case where no Byzantine modifications are introduced. ■

Proof of Corollary 1: Suppose more than one different sets of packets are used for decoding. Consider the sets in turn, denoting by s_i the number of unmodified packets in the i^{th} set that are not in any set $j < i$. For any particular values of packets in sets $j < i$, we have from Theorem 10 that at most a fraction $\left(\frac{\kappa+1}{q}\right)^{s_i}$ of decoding outcomes for set i have consistent data and hash values. Thus, the overall fraction of consistent decoding outcomes is at most $\left(\frac{\kappa+1}{q}\right)^{\sum_i s_i} = \left(\frac{\kappa+1}{q}\right)^{s'}$. ■

Chapter 6

Network coding for arbitrarily correlated sources

Having considered distributed randomized network coding for independent or linearly correlated sources in the preceding chapters, we now round out the picture by considering arbitrarily correlated sources. We find that, as in the case of independent or linearly correlated sources, a randomized network coding technique can approach optimal capacity for correlated sources, with error probability decreasing exponentially in the length of the codes and in excess multicast capacity.

This distributed randomized network coding approach effectively removes or adds data redundancy in different parts of the network depending on the available capacity. This is achieved without knowledge of the source entropy rates or network topology at interior network nodes. Compression is done simultaneously for multiple receivers in a multicast session.

6.1 Problem statement and approach

We consider linear network coding in the context of a distributed source coding problem, where compression may be required to transmit information from correlated sources over a network to one or more receivers. An example of such a problem is given in Figure 6-1.

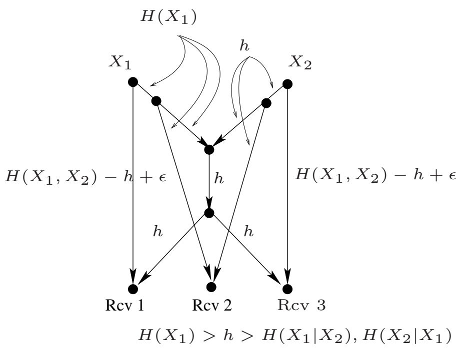


Figure 6-1: An example network with two correlated sources X_1, X_2 that can be transmitted using distributed randomized network coding. The label on each link represents the capacity of the link.

We use a distributed randomized coding approach similar to that in Chapter 4 for independent or linearly correlated sources, except that instead of scalar operations over a finite field $\mathbb{F}_{2^u}$, we employ vector operations in the binary field. In this vector coding approach, nodes other than the receiver nodes independently select random linear mappings from vectors of input bits onto vectors of output bits. The length of each vector is proportional to its corresponding link's capacity (for vectors corresponding to links) or its corresponding source's bit rate (for vectors corresponding to source processes), which are assumed to be integers[1]. An illustration is given in Figure 6-2. While scalar network codes can be specified with fewer coefficients than corresponding vector network codes of the same block length, thus requiring less overhead to specify to the receivers, the scalar coding approach of previous chapters does not generalize as easily to the case of compressible and arbitrarily correlated sources.

The vector coding model can, for given vector lengths, be brought into the scalar algebraic framework of [39] by conceptually expanding each source into multiple sources and each link into multiple links, such that each new source and link corresponds to one bit in the code vectors. We use this scalar framework to analyze the operation of interior network nodes. Note however that the linear decoding strategies of [39] do not apply when we consider compressible and arbitrarily correlated sources.

[1]As before, we can model a large class of networks and sources by choosing the time unit appropriately

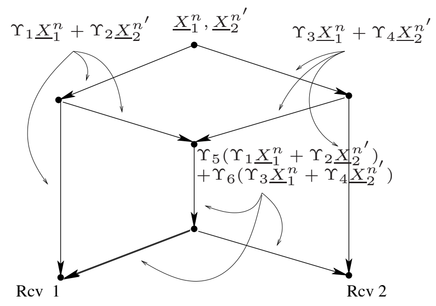


Figure 6-2: An example of distributed randomized network coding. $\underline{X}_1^n$ and $\underline{X}_2^{n'}$ are vectors of source bits being multicast to the receivers, and the matrices Υ_i are matrices of random bits. The label on each link represents the signal being carried on the link.

6.2 Main result and discussion

We consider transmission of arbitrarily correlated sources in a network by linear network coding, and show error bounds on the probability of successful (non-linear) decoding at a receiver. Analogously to Slepian and Wolf [61], we consider the problem of distributed encoding and joint decoding of two sources whose output symbols in each unit time period are drawn i.i.d. from the same joint distribution Q. The difference is that in our problem, transmission occurs across a network of intermediate nodes that perform linear transformations from their inputs to their outputs. In the special case of a network consisting of one receiver connected directly by a capacitated link to each source, this reduces to the original Slepian-Wolf problem.

An α-decoder (which may be a minimum entropy or maximum Q-probability decoder) [9] at the receiver maps a block of received signals to the corresponding minimum entropy or maximum Q-probability inputs. We derive the error probability in terms of the block length when all non-receiver nodes independently and randomly choose vector linear mappings from inputs to outputs.

The following theorem bounds the probability of successful minimum entropy or maximum a posteriori probability decoding at a receiver, for two sources X_1 and X_2 whose output values in each unit time period are drawn i.i.d. from the same joint distribution Q. We denote by r_i the bit rate of source X_i, and suppose linear coding is done in $\mathbb{F}_2$ over vectors of nr_1 and nr_2 bits from each source respectively. Let m_1

and m_2 be the minimum cut capacities between the receiver and each of the sources respectively, and let m_3 be the minimum cut capacity between the receiver and both sources. We denote by L the maximum source-receiver path length. In this chapter, all exponents and logs are taken to base 2.

Theorem 11 *For distributed randomized network coding of arbitrarily correlated sources X_1 and X_2 over an arbitrary network, the error probability is at most $\sum_{i=1}^{3} p_e^i$, where*

$$\begin{aligned}
p_e^1 &\leq \exp\Bigg\{-n \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big|m_1(1-\frac{1}{n}\log L) - H(X_1|X_2)\Big|^+\Big) \\
&\quad + 2^{2r_1+r_2}\log(n+1)\Bigg\} \\
p_e^2 &\leq \exp\Bigg\{-n \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big|m_2(1-\frac{1}{n}\log L) - H(X_2|X_1)\Big|^+\Big) \\
&\quad + 2^{r_1+2r_2}\log(n+1)\Bigg\} \\
p_e^3 &\leq \exp\Bigg\{-n \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big|m_3(1-\frac{1}{n}\log L) - H(X_1X_2)\Big|^+\Big) \\
&\quad + 2^{2r_1+2r_2}\log(n+1)\Bigg\}.
\end{aligned}$$

The proof is given in the following section.

The error exponents

$$\begin{aligned}
e^1 &= \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big|m_1(1-\frac{1}{n}\log L) - H(X_1|X_2)\Big|^+\Big) \\
e^2 &= \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big|m_2(1-\frac{1}{n}\log L) - H(X_2|X_1)\Big|^+\Big)
\end{aligned}$$

$$
\begin{aligned}
e^3 &= \min_{X_1,X_2} \Big(D(P_{X_1X_2}||Q) \\
&\quad + \Big| m_3(1 - \frac{1}{n}\log L) - H(X_1X_2) \Big|^+ \Big)
\end{aligned}
$$

generalize the Slepian-Wolf error exponents for linear coding [9]:

$$
\begin{aligned}
e^1 &= \min_{X_1,X_2} \big(D(P_{X_1X_2}||Q) + |R_1 - H(X_1|X_2)|^+ \big) \\
e^2 &= \min_{X_1,X_2} \big(D(P_{X_1X_2}||Q) + |R_2 - H(X_2|X_1)|^+ \big) \\
e^3 &= \min_{X_1,X_2} \big(D(P_{X_1X_2}||Q) + |R_1 + R_2 - H(X_1X_2)|^+ \big)
\end{aligned}
$$

where R_i is the rate of the code for X_i.

In the special case of a Slepian-Wolf source network consisting of one receiver connected directly by a capacitated link to each source, our error exponents reduce to the corresponding results for linear Slepian-Wolf coding. The latter scenario may thus be considered a degenerate case of network coding.

This randomized network coding approach carries over to any positive number of sources, though we give here a detailed treatment only of the case of two sources. Our error bounds are in terms of minimum cut capacities and maximum source-receiver path length in a network. Bounds in terms of other network parameters, e.g. the number of links upstream of a receiver, or for particular network topologies, can be obtained using similar means.

6.3 Proof

Proof of Theorem 11: Encoding in the network is represented by the transfer matrix $AG_{\mathcal{T}}$ specifying the mapping from the vector of source signals $[\ X_1 \quad X_2\ ] \in \mathbb{F}_2^{n(r_1+r_2)}$ to the vector $\mathbf{z}$ of signals on the set $\mathcal{T}$ of terminal links incident to the receiver. Our error analysis, using the method of types, is similar to that in [9]. As there, the type $P_{\mathbf{x}_i}$ of a vector $\mathbf{x}_i \in \mathbb{F}_2^{nr_i}$ is the distribution on $\mathbb{F}_2$ defined by the relative frequencies of the elements of $\mathbb{F}_2$ in $\mathbf{x}_i$, and joint types $P_{\mathbf{x}_1\mathbf{x}_2}$ are analogously defined.

The α-decoder maps a vector $\mathbf{z}$ of received signals onto a vector in $\mathbb{F}_2^{n(r_1+r_2)}$ minimizing $\alpha(P_{\mathbf{x}_1\mathbf{x}_2})$ subject to $[\ \mathbf{x}_1 \quad \mathbf{x}_2\ ]AG_{\mathcal{T}} = \mathbf{z}$. For a minimum entropy decoder, $\alpha(P_{\mathbf{x}_1\mathbf{x}_2}) \equiv H(P_{\mathbf{x}_1\mathbf{x}_2})$, while for a maximum Q-probability decoder, $\alpha(P_{\mathbf{x}_1\mathbf{x}_2}) \equiv -\log Q^n(\mathbf{x}_1\mathbf{x}_2)$. We consider three types of errors: in the first type, the decoder has the correct value for X_2 but outputs the wrong value for X_1; in the second, the decoder has the correct value for X_1 but outputs the wrong value for X_2; in the third, the decoder outputs wrong values for both X_1 and X_2. The error probability is bounded from above by the sum of the probabilities of the three types of errors, $\sum_{i=1}^{3} p_e^i$. Defining the sets of types

$$\mathcal{P}_n^i = \begin{cases} \{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \mid \tilde{X}_1 \neq X_1, \tilde{X}_2 = X_2\} & i = 1 \\ \{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \mid \tilde{X}_1 = X_1, \tilde{X}_2 \neq X_2\} & i = 2 \\ \{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \mid \tilde{X}_1 \neq X_1, \tilde{X}_2 \neq X_2\} & i = 3 \end{cases}$$

where $\tilde{X}_i \in \mathbb{F}_2^{nr_i}$, and the sets of sequences

$$\begin{aligned} \mathcal{T}_{X_1X_2} &= \{[\ \mathbf{x}_1 \quad \mathbf{x}_2\ ] \in \mathbb{F}_2^{n(r_1+r_2)} \mid P_{\mathbf{x}_1\mathbf{x}_2} = P_{X_1X_2}\} \\ \mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2) &= \{[\ \tilde{\mathbf{x}}_1 \quad \tilde{\mathbf{x}}_2\ ] \in \mathbb{F}_2^{n(r_1+r_2)} \mid \\ &\qquad\qquad P_{\tilde{\mathbf{x}}_1\tilde{\mathbf{x}}_2\mathbf{x}_1\mathbf{x}_2} = P_{\tilde{X}_1\tilde{X}_2X_1X_2}\} \end{aligned}$$

we have

$$\begin{aligned} p_e^1 \leq & \sum_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^1 : \\ \alpha(P_{\tilde{X}_1X_2}) \leq \alpha(P_{X_1X_2})}} \sum_{\substack{(\mathbf{x}_1,\mathbf{x}_2) \in \\ \mathcal{T}_{X_1X_2}}} Q^n(\mathbf{x}_1\mathbf{x}_2) \\ & \quad \Pr\left(\exists(\tilde{\mathbf{x}}_1,\tilde{\mathbf{x}}_2) \in \mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2) \text{ s.t.} [\ \mathbf{x}_1 - \tilde{\mathbf{x}}_1 \quad \underline{0}\ ]AG_{\mathcal{T}} = \underline{0}\right) \\ \leq & \sum_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^1 : \\ \alpha(P_{\tilde{X}_1X_2}) \leq \alpha(P_{X_1X_2})}} \sum_{\substack{(\mathbf{x}_1,\mathbf{x}_2) \in \\ \mathcal{T}_{X_1X_2}}} Q^n(\mathbf{x}_1\mathbf{x}_2) \\ & \quad \min\Bigg\{ \sum_{\substack{(\tilde{\mathbf{x}}_1,\tilde{\mathbf{x}}_2) \in \\ \mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2)}} \Pr\left([\ \mathbf{x}_1 - \tilde{\mathbf{x}}_1 \quad \underline{0}\ ]AG_{\mathcal{T}} = \underline{0}\right),\ 1\Bigg\} \end{aligned}$$

Similarly,

$$
\begin{aligned}
p_e^2 &\le \sum_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^2 : \\ \alpha(P_{X_1\tilde{X}_2}) \le \alpha(P_{X_1X_2})}} \sum_{\substack{(\mathbf{x}_1,\mathbf{x}_2) \in \\ \mathcal{T}_{X_1X_2}}} Q^n(\mathbf{x}_1\mathbf{x}_2) \\
&\quad \min\Bigg\{ \sum_{\substack{(\tilde{\mathbf{x}_1},\tilde{\mathbf{x}_2}) \in \\ \mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2)}} \Pr\Big([\ \underline{0} \quad \mathbf{x}_2 - \tilde{\mathbf{x}_2}\ ]AG_{\mathcal{T}} = \underline{0}\Big)\,,\ 1\Bigg\} \\
p_e^3 &\le \sum_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^3 : \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \le \alpha(P_{X_1X_2})}} \sum_{\substack{(\mathbf{x}_1,\mathbf{x}_2) \in \\ \mathcal{T}_{X_1X_2}}} Q^n(\mathbf{x}_1\mathbf{x}_2) \\
&\quad \min\Bigg\{ \sum_{\substack{(\tilde{\mathbf{x}_1},\tilde{\mathbf{x}_2}) \in \\ \mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2)}} \Pr\Big([\ \mathbf{x}_1 - \tilde{\mathbf{x}_1} \quad \mathbf{x}_2 - \tilde{\mathbf{x}_2}\ ]AG_{\mathcal{T}} = \underline{0}\Big),\ 1\Bigg\}
\end{aligned}
$$

where the probabilities are taken over realizations of the network transfer matrix $AG_{\mathcal{T}}$ corresponding to the random network code. The probabilities

$$
\begin{aligned}
p_1 &= \Pr\Big([\ \mathbf{x}_1 - \tilde{\mathbf{x}_1} \quad \underline{0}\ ]AG_{\mathcal{T}} = \underline{0}\Big) \\
p_2 &= \Pr\Big([\ \underline{0} \quad \mathbf{x}_2 - \tilde{\mathbf{x}_2}\ ]AG_{\mathcal{T}} = \underline{0}\Big) \\
p_3 &= \Pr\Big([\ \mathbf{x}_1 - \tilde{\mathbf{x}_1} \quad \mathbf{x}_2 - \tilde{\mathbf{x}_2}\ ]AG_{\mathcal{T}} = \underline{0}\Big)
\end{aligned}
$$

for nonzero $\mathbf{x}_1 - \tilde{\mathbf{x}_1}, \mathbf{x}_2 - \tilde{\mathbf{x}_2}$ can be calculated for a given network, or bounded in terms of n and parameters of the network as we will show later.

As in [9], we can apply some simple cardinality bounds

$$
\begin{aligned}
|\mathcal{P}_n^1| &< (n+1)^{2^{2r_1+r_2}} \\
|\mathcal{P}_n^2| &< (n+1)^{2^{r_1+2r_2}} \\
|\mathcal{P}_n^3| &< (n+1)^{2^{2r_1+2r_2}} \\
|\mathcal{T}_{X_1X_2}| &\le \exp\{nH(X_1X_2)\} \\
|\mathcal{T}_{\tilde{X}_1\tilde{X}_2|X_1X_2}(\mathbf{x}_1\mathbf{x}_2)| &\le \exp\{nH(\tilde{X}_1\tilde{X}_2|X_1X_2)\}
\end{aligned}
$$

and the identity

$$Q^n(\mathbf{x}_1\mathbf{x}_2) = \exp\{-n(D(P_{X_1X_2}||Q) + H(X_1X_2))\},$$
$$(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{T}_{X_1X_2} \quad (6.1)$$

to obtain

$$\begin{aligned}
p_e^1 \leq \exp\Bigg\{ - n \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^1: \\ \alpha(P_{\tilde{X}_1X_2}) \leq \alpha(P_{X_1X_2})}} & \Big(D(P_{X_1X_2}||Q) \\
& + \left| -\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_1X_2) \right|^+ \Big) \\
& + 2^{2r_1+r_2}\log(n+1) \Bigg\} \\
p_e^2 \leq \exp\Bigg\{ - n \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^2: \\ \alpha(P_{X_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2})}} & \Big(D(P_{X_1X_2}||Q) \\
& + \left| -\frac{1}{n}\log p_2 - H(\tilde{X}_2|X_1X_2) \right|^+ \Big) \\
& + 2^{r_1+2r_2}\log(n+1) \Bigg\} \\
p_e^3 \leq \exp\Bigg\{ - n \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^3: \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2})}} & \Big(D(P_{X_1X_2}||Q) \\
& + \left| -\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2|X_1X_2) \right|^+ \Big) \\
& + 2^{2r_1+2r_2}\log(n+1) \Bigg\},
\end{aligned}$$

For the minimum entropy decoder, we have

$$\alpha(P_{\tilde{X}_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2})$$

$$\Rightarrow \begin{cases} H(\tilde{X}_1|X_1X_2) \leq H(\tilde{X}_1|X_2) \leq H(X_1|X_2) \\ \qquad\qquad\qquad\qquad \text{for } X_2 = \tilde{X}_2 \\ H(\tilde{X}_2|X_1X_2) \leq H(\tilde{X}_2|X_1) \leq H(X_2|X_1) \\ \qquad\qquad\qquad\qquad \text{for } X_1 = \tilde{X}_1 \\ H(\tilde{X}_1\tilde{X}_2|X_1X_2) \leq H(\tilde{X}_1\tilde{X}_2) \leq H(X_1X_2) \end{cases},$$

which gives

$$\begin{aligned} p_e^1 \leq \exp\Bigg\{ &- n \min_{X_1X_2} \Big(D(P_{X_1X_2}||Q) \\ &+ \left| -\frac{1}{n} \log p_1 - H(X_1|X_2) \right|^+ \Big) \\ &+ 2^{2r_1+r_2} \log(n+1) \Bigg\} \end{aligned} \tag{6.2}$$

$$\begin{aligned} p_e^2 \leq \exp\Bigg\{ &- n \min_{X_1X_2} \Big(D(P_{X_1X_2}||Q) \\ &+ \left| -\frac{1}{n} \log p_2 - H(X_2|X_1) \right|^+ \Big) \\ &+ 2^{r_1+2r_2} \log(n+1) \Bigg\} \end{aligned} \tag{6.3}$$

$$\begin{aligned} p_e^3 \leq \exp\Bigg\{ &- n \min_{X_1X_2} \Big(D(P_{X_1X_2}||Q) \\ &+ \left| -\frac{1}{n} \log p_3 - H(X_1X_2) \right|^+ \Big) \\ &+ 2^{2r_1+2r_2} \log(n+1) \Bigg\}. \end{aligned} \tag{6.4}$$

We next show that these bounds also hold for the maximum Q-probability decoder, for which, from (6.1),

$$\begin{aligned} &\alpha(P_{\tilde{X}_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2}) \\ &\Rightarrow D(P_{\tilde{X}_1\tilde{X}_2}||Q) + H(\tilde{X}_1\tilde{X}_2) \\ &\quad \leq D(P_{X_1X_2}||Q) + H(X_1X_2). \end{aligned} \tag{6.5}$$

For $i = 1$, $\tilde{X}_2 = X_2$, and (6.5) gives

$$D(P_{\tilde{X}_1 X_2}||Q) + H(\tilde{X}_1|X_2) \le D(P_{X_1 X_2}||Q) + H(X_1|X_2). \tag{6.6}$$

We show that

$$\begin{aligned}
&\min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^1 : \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \le \alpha(P_{X_1X_2})}} \left(D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_1X_2) \right|^+ \right) \\
&\quad \ge \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^1 : \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \le \alpha(P_{X_1X_2})}} \left(D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2) \right|^+ \right) \\
&\quad \ge \min_{X_1X_2} \left(D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(X_1|X_2) \right|^+ \right)
\end{aligned}$$

by considering two possible cases for any $X_1, \tilde{X}_1, X_2$ satisfying (6.6):

Case 1: $-\frac{1}{n}\log p_1 - H(X_1|X_2) < 0$. Then

$$\begin{aligned}
&D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2) \right|^+ \\
&\ge D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(X_1|X_2) \right|^+ \\
&\ge \min_{X_1X_2} \left(D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(X_1|X_2) \right|^+ \right)
\end{aligned}$$

Case 2: $-\frac{1}{n}\log p_1 - H(X_1|X_2) \ge 0$. Then

$$\begin{aligned}
&D(P_{X_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2) \right|^+ \\
\ge\ & D(P_{X_1X_2}||Q) - \frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2) \\
\ge\ & D(P_{\tilde{X}_1X_2}||Q) - \frac{1}{n}\log p_1 - H(X_1|X_2) \text{ by (6.6)} \\
=\ & D(P_{\tilde{X}_1X_2}||Q) + \left| -\frac{1}{n}\log p_1 - H(X_1|X_2) \right|^+,
\end{aligned}$$

which gives

$$
\begin{aligned}
& D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2)\right|^+ \\
\geq\ & \frac{1}{2}\Big[D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_1 - H(\tilde{X}_1|X_2)\right|^+ \\
& + D(P_{\tilde{X}_1X_2}||Q) + \left|-\frac{1}{n}\log p_1 - H(X_1|X_2)\right|^+\Big] \\
\geq\ & \min_{X_1X_2}\Big(D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_1 - H(X_1|X_2)\right|^+\Big).
\end{aligned}
$$

A similar proof holds for $i = 2$.

For $i = 3$, we show that

$$
\begin{aligned}
& \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^3 : \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2})}} \Big(D(P_{X_1X_2}||Q) \\
& \qquad + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2|X_1X_2)\right|^+\Big) \\
& \geq \min_{\substack{P_{X_1\tilde{X}_1X_2\tilde{X}_2} \in \mathcal{P}_n^3 : \\ \alpha(P_{\tilde{X}_1\tilde{X}_2}) \leq \alpha(P_{X_1X_2})}} \Big(D(P_{X_1X_2}||Q) \\
& \qquad + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2)\right|^+\Big) \\
& \geq \min_{X_1X_2}\Big(D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+\Big)
\end{aligned}
$$

by considering two possible cases for any $X_1, \tilde{X}_1, X_2, \tilde{X}_2$ satisfying (6.5):

Case 1: $-\frac{1}{n}\log p_3 - H(X_1X_2) < 0$. Then

$$
\begin{aligned}
& D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2)\right|^+ \\
& \geq D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+ \\
& \geq \min_{X_1X_2}\Big(D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+\Big)
\end{aligned}
$$

Case 2: $-\frac{1}{n}\log p_3 - H(X_1X_2) \geq 0$. Then

$$\begin{aligned}
& D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2)\right|^+ \\
\geq\ & D(P_{X_1X_2}||Q) - \frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2) \\
\geq\ & D(P_{\tilde{X}_1\tilde{X}_2}||Q) - \frac{1}{n}\log p_3 - H(X_1X_2) \text{ by (6.5)} \\
=\ & D(P_{\tilde{X}_1\tilde{X}_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+
\end{aligned}$$

which gives

$$\begin{aligned}
& D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2)\right|^+ \\
\geq\ & \frac{1}{2}\Big[D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(\tilde{X}_1\tilde{X}_2)\right|^+ \\
& +D(P_{\tilde{X}_1\tilde{X}_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+\Big] \\
\geq\ & \min_{X_1X_2}\left(D(P_{X_1X_2}||Q) + \left|-\frac{1}{n}\log p_3 - H(X_1X_2)\right|^+\right).
\end{aligned}$$

Next we bound the probabilities p_i in terms of n and the network parameters $m_i, i = 1, 2$, the minimum cut capacity between the receiver and source X_i, m_3, the minimum cut capacity between the receiver and both sources, and L, the maximum source-receiver path length. Let $\mathcal{G}_1, \mathcal{G}_2$, be subgraphs of graph $\mathcal{G}$ consisting of all links downstream of sources 1 and 2 respectively, and let $\mathcal{G}_3$ be equal to $\mathcal{G}$. It follows from our algebraic coding model that in a random linear network code over an arbitrary network, any link which has at least one nonzero incoming signal carries the zero signal with probability $\frac{1}{2^{nc}}$, where c is the capacity of the link. This is the same as the probability that a pair of distinct values for the link's inputs are mapped to the same output on the link.

For a given pair of distinct source values, let E_l be the event that the corresponding inputs to link l are distinct, but the corresponding values on l are the same. Let $E(\tilde{\mathcal{G}})$ be the event that E_l occurs for some link l on every source-receiver path in graph $\tilde{\mathcal{G}}$. p_i is then equal to the probability of event $E(\mathcal{G}_i)$.

Let $\mathcal{G}'_i, i = 1, 2, 3$ be the graph consisting of m_i node-disjoint paths, each consisting of L links each of unit capacity. We show by induction on m_i that p_i is upper bounded by the probability of event $E(\mathcal{G}'_i)$.

We let $\tilde{\mathcal{G}}$ be the graphs $\mathcal{G}_i, \mathcal{G}'_i, i = 1, 2, 3$ in turn, and consider any particular source-receiver path $P_{\tilde{\mathcal{G}}}$ in $\tilde{\mathcal{G}}$. We distinguish two cases:

Case 1: E_l does not occur for any of the links l on the path $P_{\tilde{\mathcal{G}}}$. In this case the event $E(\tilde{\mathcal{G}})$ occurs with probability 0.

Case 2: There exists some link $\hat{l}$ on the path $P_{\tilde{\mathcal{G}}}$ for which E_l occurs.

Thus, we have $\Pr(E(\tilde{\mathcal{G}})) = \Pr(\text{case 2}) \Pr(E(\tilde{\mathcal{G}})|\text{case 2})$. Since $P_{\mathcal{G}'_i}$ has at least as many links as $P_{\mathcal{G}_i}$, $\Pr(\text{case 2 for } \mathcal{G}'_i) \geq \Pr(\text{case 2 for } \mathcal{G}_i)$. Therefore, if we can show that $\Pr(E(\mathcal{G}'_i)|\text{case 2}) \geq \Pr(E(\mathcal{G}_i)|\text{case 2})$, the induction hypothesis $\Pr(E(\mathcal{G}'_i)) \geq \Pr(E(\mathcal{G}_i))$ follows.

For $m_i = 1$, the hypothesis is true since $\Pr(E(\mathcal{G}'_i)|\text{case 2}) = 1$. For $m_i > 1$, in case 2, removing the link $\hat{l}$ leaves, for $\mathcal{G}'_i$, the effective equivalent of a graph consisting of $m_i - 1$ node-disjoint length-L paths, and, for $\mathcal{G}_i$, a graph of minimum cut at least $m_i - 1$. The result follows from applying the induction hypothesis to the resulting graphs.

Thus, $\Pr(E(\mathcal{G}'_i))$ gives an upper bound on probability p_i:

$$\begin{aligned} p_i &\leq \left(1 - (1 - \frac{1}{2^n})^L\right)^{m_i} \\ &\leq \left(\frac{L}{2^n}\right)^{m_i}. \end{aligned}$$

Substituting this into the error bounds (6.2)-(6.4) gives the desired results. ■

Chapter 7

A coding view of network management

In this chapter, we move away from the randomized coding approach of previous chapters and consider a theoretical application of network coding. We develop an information theoretic framework, based on network coding, for quantifying and bounding fundamental network management requirements for link failure recovery.

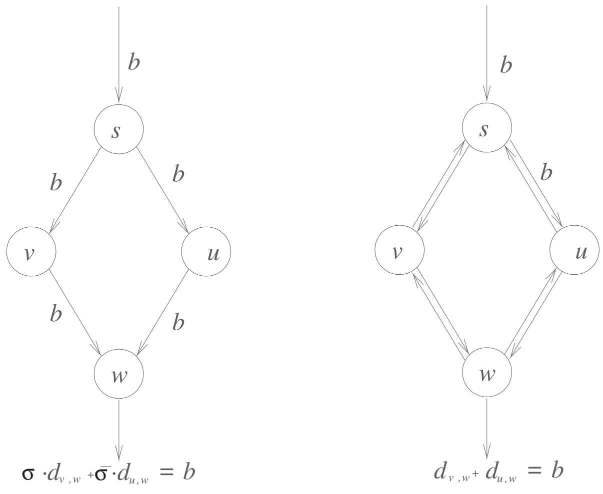
7.1 Background

Network management for protection and restoration in the case of failures has generally been considered in an *ad hoc* manner, within the context of specific schemes. These schemes are predominantly routing schemes, and the use of network coding, which in contrast to routing allows a network node to form outgoing data from incoming data in an arbitrary fashion and possibly involving network management signals, to describe them may at first appear superfluous. However, it will turn out that enlarging the set of allowed operations at network nodes not only opens new and fruitful ways to protect networks, but the framework also naturally integrates traditional, well known solutions to the problem of robust networks.

To illustrate this point, we consider two of the most common means of providing network recovery for non-ergodic failures, showing how a coding framework offers a simple and systematic approach to describing such recovery schemes. Within preplanned methods for network recovery, generally termed protection, we may distinguish between path and link or node protection. Path protection refers to recovery applied to connections following a particular path across a network. Link or node restoration refers to recovery of all the traffic across a failed link or node, respectively. An overview of restoration and recovery can be found in [55, 56]. Path restoration may be itself subdivided into two different types: live (dual-fed) back-up and event-triggered back-up. In the first case, two live flows, a primary and a back-up, are transmitted. The two flows are link-disjoint if we seek to protect against link failure, or node-disjoint (except for the end nodes) if we seek to protect against node failure. Recovery is extremely fast, requiring action only from the receiving node, but back-up capacity is not shared among connections. In the second case, event-triggered path protection, the back-up path is only activated when a failure occurs on a link or node along the primary path. Backup capacity can be shared among different paths [64], thus improving capacity utilization for back-up channels and allowing for judicious planning [2, 24, 22, 23, 20, 58, 62, 18, 51]. However, recovery involves coordination between the sender and receiver after a failure event and requires action from nodes along the back-up path.

Figure 7-1 illustrates our discussion, which uses a simple four-node ring as its basis. We have a single sender s transmitting data b to a single receiver w.

The simplest scheme to consider is live path protection, shown in Figure 7-1.a. The primary path is $s \rightarrow v \rightarrow w$. At the receiver, the only network supervisory signal required is a signal indicating whether or not the primary path is live. The supervisory signal is denoted by σ, where σ is 1 if the primary path has had no failures and is 0 otherwise. Let $d_{i,j}$ denote the data being sent along directed link (i, j). In order to express the protection mechanism in the framework of network coding, we need to exhibit the rules by which outgoing data streams are formed from incoming data and potentially network management signals. For links (s, u), (u, w), (s, v) and (v, w), the rules are trivial in that the outgoing data equals the incoming data, which is b. The behavior, or code, at w is shown in Figure 7-1.a.



(a) (b)

Figure 7-1: Ring network illustrating path protection in (a) and link protection in (b).

For failure-induced path protection, the sender knows σ. The code is similar to the one in Figure 7-1.a. The links in the backup path carry the same signal as for the live path, but multiplied by $\overline{\sigma}$, which means that nothing is carried except in the case of failure. The links in the primary path see their data multiplied by σ. The receiver need not have knowledge of σ. It simply outputs $d_{u,w} + d_{v,w}$.

Link recovery is illustrated in Figure 7-1.b. We have primary links, which are the links in the clockwise direction and backup (secondary) links, which are the links in the counterclockwise direction. The supervisory signal $\sigma_{i,j}$ is 1 if the primary link from i to node j has not failed and is 0 otherwise. Thus, the supervisory signal is no longer associated with a full path, but rather with a link, regardless of what routes, if any, traverse that link. Consider, in our ring, any three consecutive nodes k, i, h. These nodes and their links are shown in Figure 7-2. The thick lines represent primary links, which transmit information when no link failures occur, and the thin lines represent secondary links, which transmit information when a failure occurs. The code for the primary link (i, h) emanating from i is $d_{i,h} = d_{k,i} + d_{h,i}$ (where (k, i) is the primary link into i and (h, i) is the secondary link into i) except when $i = s$,

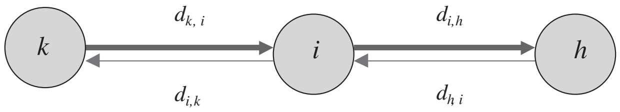


Figure 7-2: Three nodes and their primary (thick) and backup (thin) links.

for which it is the incoming signal b. For the secondary link emanating from i, the code is $d_{i,k} = d_{h,i}.\sigma_{i,h} + d_{i,h}.\overline{\sigma}_{i,h}$. The output at node w, as shown in Figure 7-1.b, is the sum of the signals on its incoming primary and incoming secondary links. Thus, by specifying the local behaviors of nodes, the concept of link recovery fits naturally in the framework of network coding.

The example above illustrates how network coding can provide an efficient vehicle for formalizing traditional recovery problems. Similar techniques can be applied to describe the operation of a wide array of recovery techniques over complex topologies, for instance by using ring covers [63, 21, 15, 14] or generalized loop-back [48]. Our goal, however, is not to merely translate known recovery approaches and their related network management mechanisms into a network coding setting. Instead, we seek to use a coding approach over networks to obtain fundamental results concerning network management.

7.2 Problem statement

We may formulate a basic general form of the network management problem as shown in the block diagram in Figure 7-3. A network is modelled as a mapping from a set of inputs $\lambda \in \Lambda$ to a set of outputs $\eta \in \Upsilon$. This mapping $\mu_{s,c} : \Lambda \rightarrow \Upsilon$ depends on the state s of the network: for instance, network outputs are affected by link or node failures in the network. The mapping can also be affected by management signals $c \in \mathcal{C}$ that change the behavior of network nodes: for instance, causing a node to switch between using different output links. Different management signals can be applied appropriately based on observations $o(s) \in \mathcal{O}(s)$ of the network state. We consider the network management problem of determining the minimum cardinality

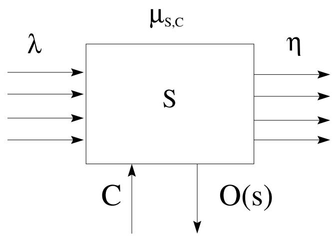


Figure 7-3: General network management problem.

of the set $\mathcal{C}$ of management signals needed, given a set of possible network states and a set of required input-output connections that must be maintained across these states.

The particular problem we focus on in this chapter is network management for link failures, for which various existing recovery schemes have been described earlier. What these schemes have in common is a need for detecting failures, and directing network nodes to respond appropriately.

While failure detection is itself an important issue, it is the latter component of management overhead, that of directing recovery behavior, that we seek here to understand and quantify in a fundamental way. This work is an attempt to start developing a theory of network management for non-ergodic failures. Our aim is to examine network management in a way that is abstracted from specific implementations, while fully recognizing that implementation issues are interesting, numerous and difficult. Network coding gives us a framework for considering this. The very general concept of network behavior as a code provides a fundamental way to quantify essential management information as that needed to switch among different codes (behaviors) for different failure scenarios.

We consider two formulations for quantifying network management. In the first, a *centralized* formulation, the management requirement is taken as the logarithm of the number of codes that the network switches among. In an alternative *node-based* formulation, the management requirement is defined as the sum over all nodes of the logarithm of the number of behaviors for each node. For each of these formulations, we analyze network management requirements for receiver-based recovery, which in-

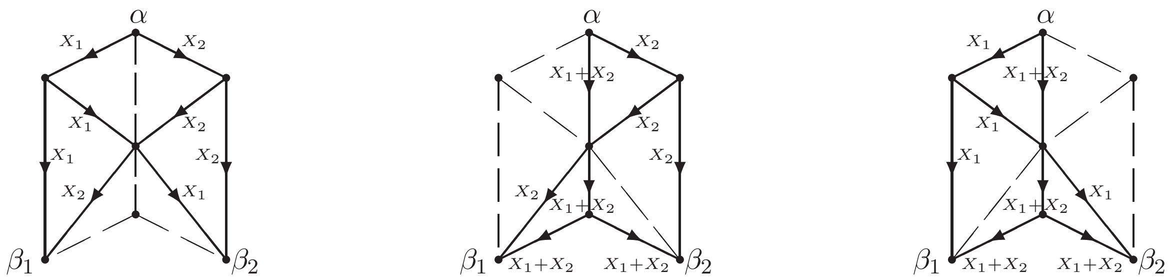


Figure 7-4: An example of a receiver-based recovery scheme. Each diagram corresponds to a code valid for failure of any of the links represented by dashed lines. The only nodes that alter their input-output relations across the three codes are the receiver nodes β_1 and β_2.

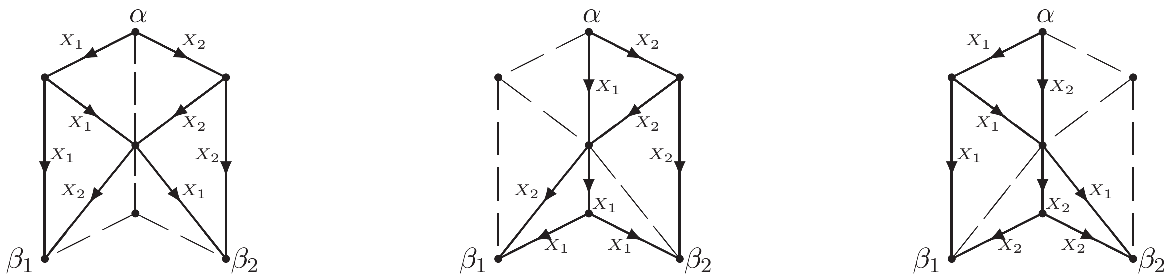


Figure 7-5: An example of a network-wide recovery scheme. Each diagram gives a code which is valid for failure of any of the links represented by dashed lines.

volves only receiver nodes, and for *network-wide* recovery, which may involve any combination of interior nodes and receiver nodes.

As an illustration of some key concepts, consider the simple example network in Figures 7-4 and 7-5, in which a source node α simultaneously sends processes X_1 and X_2 to two receiver nodes β_1 and β_2. These connections are recoverable under failure of any one link in the network. One possible set of codes forming a receiver-based recovery scheme is shown in Figure 7-4, and a possible set of codes forming a network-wide scheme is given in Figure 7-5. For this example, routing and replication are sufficient for network-wide recovery, while coding is needed for receiver-based recovery. Here linear coding is used, i.e. outputs from a node are linear combinations of the inputs to that node.

For this example it so happens that the minimum centralized management requirement is log(3) for both receiver-based and network wide recovery, but we shall see that in some cases the centralized management requirements for receiver-based and network wide recovery can differ.

Considering the node-based network management formulation, the receiver-based scheme of Figure 7-4 has the receiver nodes switching among three codes each, so the associated node-based management requirement is $2\log(3) = 3.17$. The network-wide scheme of Figure 7-5 has the source node switching among three codes, while the receiver nodes switch between two codes each, for a node-based management requirement of $\log(3) + 2\log(2) = 3.58$.

In this chapter we focus primarily on management requirements for failure of individual unit capacity components in delay-free acyclic networks. Our analysis is based primarily on the linear coding model of Chapter 2. In addition, we consider *nonlinear receiver-based* schemes, where the interior nodes' outputs are static linear functions of their inputs as before, but the output processes $Z(\beta, i)$ at a receiver node β may be nonlinear functions of the signals on its terminal links.

We assume that when a link fails, it is effectively removed from the network, or equivalently, that a zero signal is observed on that link. An alternative is to treat signals on failed links as undetermined, which, as discussed in Section 7.4.2, restricts the class of recovery codes that can be used. For the linear coding matrices described above, failure of link h corresponds to setting to zero the h^{th} column of matrices A, B and F, and the h^{th} row of F. A recovery code (A, F, B) is said to *cover* (failure of) link h if all receiver nodes are able to reconstruct the same output processes in the same order as before the failure.

7.3 Main results

Our main results provide, for centralized network management information bits necessary to achieve recovery using linear codes from all single link failures, lower bounds for arbitrary connections and upper bounds for multi-transmitter multicast connections. For the node-based formulation, we are able to show that the minimum node-based requirement for failures of links adjacent to a single receiver is achieved with receiver-based schemes. We have not determined if this holds in general for all single

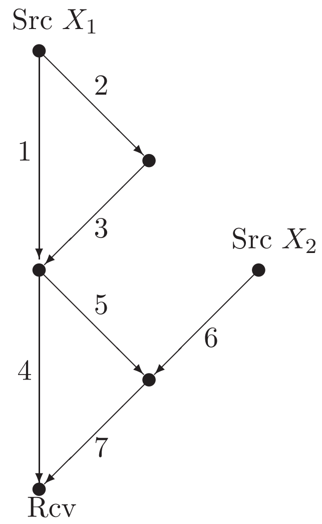


Figure 7-6: Example illustrating integral links and recoverable link failures. Sources X_1 and X_2 are required to be concurrently transmitted to the receiver. Links 1, 2 and 3 are integral, and failure of any one of them is recoverable. Links 4, 6 and 7 are integral, but their failures are not recoverable. Link 5 is not integral, but its failure is recoverable.

link failures. The proofs of our results are given in the following sections.

Our first result shows the need for network management when linear codes are used. We call a link h *integral* if it satisfies the property that there exists some subgraph of the network containing h that supports the set of source-receiver connections if and only if h has not failed. An example illustrating this definition is given in Figure 7-6.

Theorem 12 (Need for network management) *Consider any network connection problem with at least one integral link whose failure is recoverable. Then there is no single linear code (A, F, B) that can cover the no-failure scenario and all recoverable failures for this problem.* □

Although a solution with static A and F matrices always exists for any recoverable set of failures in a multicast scenario [38], in such cases the receiver code B must change. On the other hand, if we allow for non-linear processing at the receivers, in some instances this allows for unchanged network behavior over all recoverable failures.

Theorems 13-15 below give bounds on the number of codes needed for link failure recovery, in various network connection problems where all single link failures are recoverable. These bounds translate directly into bounds on the centralized network management requirement, by taking the logarithm of the number of codes. Some of these bounds are tight, in that for any values of the parameters in terms of which the bounds are given, there are examples for which these bounds are met with equality.

The bounds are given in terms of the following parameters:

- r, the number of source processes transmitted in the network;
- m, the number of links in a minimum cut between the source nodes and receiver nodes;
- $d = |\mathcal{D}|$, the number of receiver nodes;
- t_β, the number of terminal links of a receiver β;
- $t_{\min} = \min_{\beta \in \mathcal{D}} t_\beta$, the minimum number of terminal links among all receivers.

Note that our bounds do not depend on the total number of links in the network.

Theorem 13 (General lower bound for linear recovery) *For the general case, tight lower bounds on the number of linear codes for the no-failure scenario and all single link failures are:*

receiver-based	$\left\lceil \frac{m}{m-r} \right\rceil$
network-wide	$\left\lceil \frac{m+1}{m-r+1} \right\rceil$

□

Theorem 14 (Upper bounds for linear recovery)

a. For the **single-receiver** *case, tight upper bounds on the number of linear codes needed for the no-failure case and* **all single link failures** *are:*

receiver-based	$\begin{cases} r+1 & \textit{for } r=1 \;\; \textit{or} \;\; m-1 \\ r & \textit{for } 2 \le r \le m-2 \end{cases}$
network-wide	$\begin{cases} r+1 & \textit{for } r=1,\; r=2=m-1 \\ r & \textit{for } r=2 \le m-2, \\ & \qquad r=3,\; r=m-1 \ge 3 \\ r-1 & \textit{for } 4 \le r \le m-2 \end{cases}$

b. For the **multicast case with** $d \ge 2$ **receivers***, an upper bound on the number of linear codes for the no-failure scenario and* **all single link failures** *is*

$$(r^2+2)(r+1)^{d-2}.$$

c. For the **non-multicast case***, an upper bound on the number of linear codes for the no-failure scenario and all* **single terminal link failures** *is given by*

$$\sum_{\beta\,:\,t_\beta \le r} t_\beta + \sum_{\beta\,:\,t_\beta \ge r+1} (r-1)$$

where the sums are taken over receiver nodes $\beta \in \mathcal{D}$.

□

Network-wide schemes are more general than receiver-based schemes. The additional flexibility of network-wide schemes allows for smaller centralized network management requirements than receiver-based schemes in some cases, though the differences in bounds that we have found are not large. Figure 7-7 gives a plot of how the bounds look for a single-receiver network with a minimum cut size m of 8.

Our lower bounds for the general case and our upper bounds for the single-receiver case are tight. Establishing tight upper bounds for the general case is an area of further research.

Up to this point we have been considering linear codes in which the outputs at all nodes are linear functions of their inputs. If the restriction on linear processing at the receivers is relaxed, there are network connection problems for which no network management is needed. For this case, we have the following bounds:

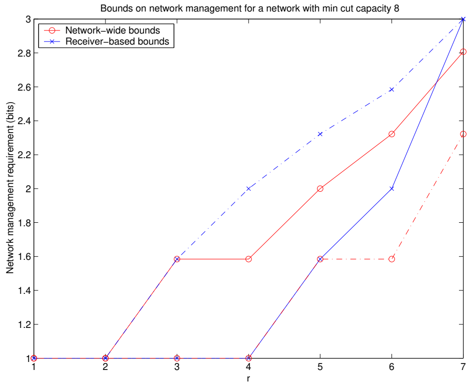


Figure 7-7: Plot of tight upper and lower bounds for centralized network management in single-receiver networks with fixed minimum cut size $m = 8$ and arbitrary numbers of links. The parameter r denotes the number of source processes transmitted in the network.

Theorem 15 (Nonlinear receiver-based recovery) *For a recovery scheme in which linear coding occurs at interior nodes but nonlinear decoding may be employed at receiver nodes, tight bounds on the number of receiver-based codes for the no-failure scenario and single terminal link failures are:*

lower bound	*upper bound*
$\begin{cases} r & \textit{for } 1 < r = t_{\min} - 1 \\ 1 & \textit{for } r = 1 \textit{ or } r \leq t_{\min} - 2 \end{cases}$	r

$\square$

Related work by Cai and Yeung [5] gives bounds on the sizes of information sources that can be transmitted through a given network with error-correcting network codes.

We have seen that the centralized management requirement may be less for network-wide schemes than for receiver-based schemes in some cases. Unlike the centralized formulation, the node-based formulation imputes higher management overhead to recovery schemes that involve more nodes, giving rise to the following result:

Theorem 16 (Node-based formulation) *For linear coding in the single-receiver case, the minimum node-based management requirement for terminal link failures and the no-failure scenario is achieved with receiver-based schemes.* $\square$

This does not however hold for the multi-receiver case. A counter-example is shown in Figure 7-8. Here, the source multicasts one process to two receivers. Linear receiver-based recovery for terminal link failures requires each of the two receivers to switch between two codes, whereas network-wide recovery allows for recovery with only the source node switching between two codes.

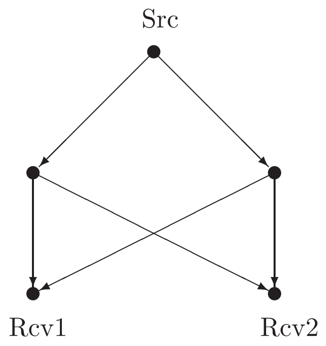


Figure 7-8: Counter-example showing that Theorem 16 does not hold for the multi-receiver case.

7.4 Detailed development, ancillary results, and proofs

7.4.1 Mathematical model

A linear network code is specified by a triple of matrices A, F and B, defined in Chapter 2. A code (A, F, B) is equivalently specified by the triple (A, G, B), where $G = (I - F)^{-1}$. A pair (A, F), or (A, G), is called an *interior code*.

We use the following notation in this chapter:

- $M(i, j)$ denotes the $(i, j)^{th}$ entry of a matrix M

- $\underline{c}_j$ and $\underline{b}_j$ denote column j of AG and B respectively. We call the column vector $\underline{c}_j$ corresponding to a link j the *signal vector* carried by j.

- $A_{\mathcal{K}}$, $G_{\mathcal{K}}$ and $B_{\beta\mathcal{K}}$ denote the submatrix of A, G, and B_β respectively consisting of columns that correspond to links in set $\mathcal{K}$.

- G^h, $G^h_{\mathcal{K}}$ and $\underline{c}^h_j$ are the altered values of G, $G_{\mathcal{K}}$ and $\underline{c}_j$ respectively resulting from failure of link h.

- $G^{\mathcal{H}}$, $G^{\mathcal{H}}_{\mathcal{K}}$ and $\underline{c}^{\mathcal{H}}_j$ are the altered values of G, $G_{\mathcal{K}}$ and $\underline{c}_j$ respectively under the combined failure of links in set $\mathcal{H}$.

- $\mathcal{T}_\beta$ is the set of terminal links of receiver β.

- $\mathcal{T}^h_\beta$ is the set of terminal links of receiver β that are downstream of link h. If there is a directed path from a link or node to another, the former is said to be *upstream* of the latter, and the latter *downstream* of the former.

In the general case, each receiver β requires a subset $\mathcal{X}_\beta$ of the set of source processes. A code (A, G, B) is *valid* if for all receivers β, $AGB_\beta^T = \left[\underline{e}_{i_1^\beta}|\dots|\underline{e}_{i^\beta_{|\mathcal{X}_\beta|}}\right]$, where $(i_1^\beta, \dots, i^\beta_{|\mathcal{X}_\beta|})$ is a particular permutation of $(1, \dots, |\mathcal{X}_\beta|)$, and $\underline{e}_i$ is the unit column vector whose only nonzero entry is in the i^{th} position[1]. In the single-receiver and multicast cases, we choose the same ordering for input and output processes, so this condition becomes $AGB_\beta^T = I \ \forall \ \beta$. An interior code (A, G) is called *valid* for the network connection problem if there exists some B for which (A, G, B) is a valid code for the problem.

The overall transfer matrix after failure of link h is $AI^hG^h(BI^h)^T = AG^hB^T$, where $I^h = I - \delta_{hh}$ is the identity matrix with a zero in the $(h, h)^{th}$ position, $F^h = I^hFI^h$, and $G^h = I^h + F^h + (F^h)^2 + \dots = I^h\left(I - FI^h\right)^{-1} = \left(I - I^hF\right)^{-1}I^h$. If failure of link h is recoverable, there exists some (A', G', B') such that for all $\beta \in \mathcal{D}$, $A'G'^hB_\beta'^T = \left[\underline{e}_{i_1^\beta}|\dots|\underline{e}_{i^\beta_{|\mathcal{X}_\beta|}}\right]$ where $\mathcal{X}_\beta = \{i_1^\beta, \dots, i^\beta_{|\mathcal{X}_\beta|}\}$.

In receiver-based recovery, only B changes, while in network-wide recovery, any combination of A, F and B may change.

[1]each receiver is required to correctly identify the processes and output them in a consistent order

7.4.2 Codes for different scenarios

As a first step in analyzing how many codes are needed for the various scenarios of no failures and individual link failures, we characterize codes that can cover multiple scenarios.

Lemma 6 (Codes covering multiple scenarios)

1. If code (A, G, B) covers the no-failure scenario and failure of link h, then

$$\underline{c}_h \sum_{j \in \mathcal{T}_\beta^h} G(h, j) \underline{b}_j^T = \mathbf{0} \ \forall \beta \in \mathcal{D},$$

where $\mathbf{0}$ is the $r \times r$ zero matrix.

2. If code (A, G, B) covers failures of links h and k, then $\forall \beta \in \mathcal{D}$, either

$$\begin{array}{ll} \textit{(a)} & \underline{c}_h \sum_{j \in \mathcal{T}_\beta^h} G(h, j) \underline{b}_j^T = \mathbf{0} \\ \quad \textit{and} & \underline{c}_k \sum_{j \in \mathcal{T}_\beta^k} G(k, j) \underline{b}_j^T = \mathbf{0} \\ \textit{or} & \\ \textit{(b)} & \gamma_{h,k} \sum_{j \in \mathcal{T}_\beta^h} G(h, j) \underline{b}_j^T = \sum_{j \in \mathcal{T}_\beta^k} G(k, j) \underline{b}_j^T \neq \mathbf{0} \\ \quad \textit{and} & \underline{c}_h = \gamma_{h,k} \underline{c}_k \neq \mathbf{0} \\ & \quad \textit{where } \gamma_{h,k} \in \mathbb{F}_{2^u} \end{array}$$

Proof: A code (A, G, B) that covers the no-failure scenario and failure of a link h satisfies, $\forall \beta \in \mathcal{D}$,

$$\begin{aligned} \mathbf{0} &= AGB_\beta^T - AG^h B_\beta^T \\ &= AG_{\mathcal{T}_\beta} B_{\beta \mathcal{T}_\beta}^T - AG_{\mathcal{T}_\beta}^h B_{\beta \mathcal{T}_\beta}^T \\ &= \sum_{j \in \mathcal{T}_\beta} (\underline{c}_j - \underline{c}_j^h) \underline{b}_j^T \\ &= \underline{c}_h \sum_{j \in \mathcal{T}_\beta^h} G(h, j) \underline{b}_j^T \end{aligned}$$

since $G(h, j)$ can be nonzero only for terminal links j that are downstream of link h.

A code (A, G, B) which covers failures of links h and k satisfies, $\forall\, \beta \in \mathcal{D}$,

$$\begin{aligned} & AG^h_{\mathcal{T}_\beta} B_{\beta}{}^T_{\mathcal{T}_\beta} = AG^k_{\mathcal{T}_\beta} B_{\beta}{}^T_{\mathcal{T}_\beta} \\ \Rightarrow\ & \underline{c}_h \sum_{j \in \mathcal{T}^h_\beta} G(h,j) \underline{b}_j^T = \underline{c}_k \sum_{j \in \mathcal{T}^k_\beta} G(k,j) \underline{b}_j^T \end{aligned}$$

Either both sides are equal to $\mathbf{0}$, or else vectors $\underline{c}_h$ and $\underline{c}_k$ which respectively span the column spaces of the left and right hand side expressions are multiples of each other, i.e. $\underline{c}_h = \gamma_{h,k} \underline{c}_k$, and vectors $\sum_{j \in \mathcal{T}^h_\beta} G(h,j) \underline{b}_j^T$ and $\sum_{j \in \mathcal{T}^k_\beta} G(k,j) \underline{b}_j^T$ which respectively span the row spaces of the left and right hand side expressions satisfy $\gamma_{h,k} \sum_{j \in \mathcal{T}^h_\beta} G(h,j) \underline{b}_j^T = \sum_{j \in \mathcal{T}^k_\beta} G(k,j) \underline{b}_j^T$. ∎

An intuitive interpretation of this lemma is provided by considering a simple characterization of codes relative to a given link as follows. A code (A, G, B) is termed *active* for a receiver β in a link h if $AG^h B_\beta^T$ is affected by the value on link h, i.e. $\underline{c}_h \sum_{j \in \mathcal{T}^h_\beta} G(h,j) \underline{b}_{\beta_j}^T \neq \mathbf{0}$. A code is *active* in a link h if it is active in h for some receiver β. Otherwise, the code is *non-active* in h. For a code which is non-active in a link h, the value on h could be set to zero (by upstream links ceasing to transmit on the link), cancelled out, or disregarded by the receivers.

By Part 1 of Lemma 6, a code which covers the no-failure scenario as well as one or more single link failures must be non-active in those links. By Part 2 of Lemma 6, a code which covers failures of two or more single links is, for each receiver, either non-active in all of them (case a) or active in all of them (case b). In the latter case, those links carry signals that are multiples of each other. We term a code *active* if it is active in those links whose failures it covers, and *non-active* otherwise. If signals on failed links are undetermined, then consideration must be restricted to non-active codes.

These expressions simplify considerably for terminal links as follows:

Corollary 2

1. If code (A, G, B) covers the no-failure scenario and failure of terminal link h, then $\underline{c}_h \underline{b}_h^T = \mathbf{0}$.

2. If (A, G, B) covers failures of two terminal links h and k, then either

$$\text{(a)} \quad \underline{c}_h \underline{b}_h^T = \mathbf{0} \quad \textit{and} \quad \underline{c}_k \underline{b}_k^T = \mathbf{0}$$

or

(b) h and k are terminal links of the same receiver β,

$$\gamma_{h,k} \underline{b}_h^T = \underline{b}_k^T \neq \mathbf{0} \quad \textit{and} \quad \underline{c}_h = \gamma_{h,k} \underline{c}_k \neq \mathbf{0}$$

where $\gamma_{h,k} \in \mathbb{F}_{2^u}$

□

Proof of Theorem 12: Consider an integral link h whose failure is recoverable, and a subgraph $\mathcal{G}'$ on which the set of source-receiver connections is feasible if and only if h has not failed. $\mathcal{G}'$ does not include all links, otherwise failure of h would not be recoverable. Then the set of links not in $\mathcal{G}'$, together with h, forms a set $\mathcal{H}$ of two or more links whose individual failures are recoverable but whose combined failures are not. By Lemma 6, a code which covers the no-failure scenario and failure of a link k is non-active in k. However, a code which is non-active in all the links in $\mathcal{H}$ is not valid. Thus, no single code can cover the no-failure scenario as well as failures of all individual links in $\mathcal{H}$. ■

7.4.3 Bounds on linear network management requirement

Single receiver analysis

Let $\mathcal{M}$ be a set of links on a minimum capacity cut between the sources and the receiver[2], where $|\mathcal{M}| = m$, and let $\mathcal{J}$ be the set of links comprising links in $\mathcal{M}$ as well as links between nodes upstream of $\mathcal{M}$.

We define the $r \times |\mathcal{J}|$ matrix $Q = (q_{i,j})$ and the $|\mathcal{J}| \times |\mathcal{J}|$ matrices $D = (d_{l,j})$ and $J = (I - D)^{-1}$, which are analogous to A, F and G respectively, but which specify only signals on links in $\mathcal{J}$. We refer to a pair (Q, J) as a *partial interior code.* $q_{i,j}$

[2]a partition of the network nodes into a set containing the sources, and another set containing the receiver, such that the number of directed links from the first set to the second is minimized

and $d_{l,j}$ (which correspond exactly with $a_{i,j}$ and $f_{l,j}$ respectively for $l, j \in \mathcal{J}$) are the coefficients of the linear combination of source signals X_i and signals on incident links l that appear on link j:

$$Y(j) = \sum_{\{i\,:\,X_i \text{ generated at } v\}} q_{i,j} X_i + \sum_{\{l\;:\;\text{head}(l)=v\}} d_{l,j} Y(l)$$

The partial interior code corresponding to given A and G matrices is given by $Q = A_{\mathcal{J}}$ and $J = G_{\mathcal{J}\times\mathcal{J}}$, the submatrix of G consisting of entries from rows and columns that correspond to links in $\mathcal{J}$. If We also define $J_{\mathcal{K}}$ to be the submatrix of J consisting of columns that correspond to links in $\mathcal{K}$.

For a minimum capacity cut $\mathcal{M}$, there exists a set of link-disjoint paths $\{P_k \mid k \in \mathcal{M}\}$, where P_k connects link k in $\mathcal{M}$ to the receiver. A partial interior code (Q, J) can be *extended* to an interior code (A, G), where $A_{\mathcal{J}} = Q$ and $G_{\mathcal{J}\times\mathcal{J}} = J$, by having each link $k \in \mathcal{M}$ transmit its signal only along the path P_k, i.e. $f_{l_1,l_2} = 0\,\forall\, l_1 \in P_k, l_2 \notin P_k$. The corresponding (A, G) is called the *extension* of (Q, J).

Lemma 7 *If failure of some link in $\mathcal{J}$ is recoverable, recovery can be achieved with a code in which no link in $\mathcal{M}$ feeds into another.*

Proof: If failure of some link $l \in \mathcal{J}$ is recoverable, then there exists a partial interior code (Q, J) in which $QJ^l_{\mathcal{M}}$ has full rank. Having one link in $\mathcal{M}$ feed into another only adds a multiple of one column of $QJ_{\mathcal{M}}$ to another, which does not increase its rank. Thus, the extension of (Q, J) is a valid code covering failure of l, with the property that no link in $\mathcal{M}$ feeds into another. ■

Let us call the original network connection problem Π, and define a related connection problem Π' on a network with

- sources and nodes corresponding exactly to those in the original network that are upstream of $\mathcal{M}$,
- links corresponding to those of the original network originating at nodes upstream of $\mathcal{M}$,

- a single receiver node β' whose terminal links h' correspond to links h in $\mathcal{M}$, with $\text{tail}(h') = \text{tail}(h)$.

An example illustrating this is given in Figure 7-9.

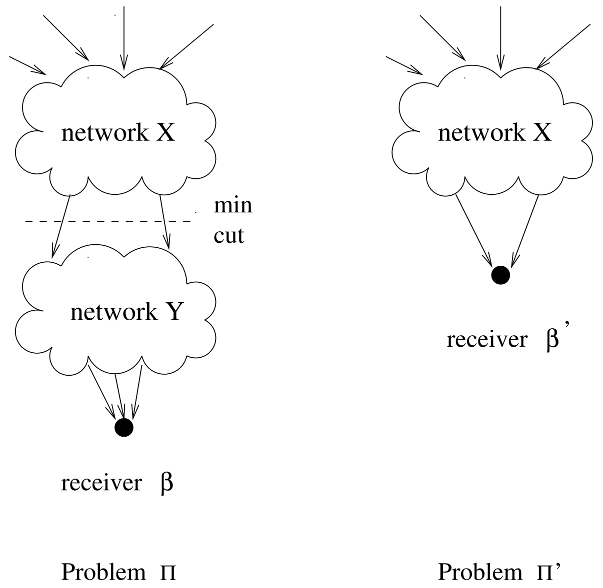


Figure 7-9: Example illustrating the definition of a related connection problem Π' from an original problem Π.

Corollary 3 *If failure of some link in $\mathcal{J}$ is recoverable in problem Π, then failure of the corresponding link in Π' is recoverable.*

The following lemma relates codes for terminal link failures in problem Π' to codes for failures of links in $\mathcal{M}$ in problem Π.

Lemma 8 *Let (Q, J) be a partial interior code in which no link in $\mathcal{M}$ feeds into another. If there exists an $r \times m$ matrix L such that $QJ_{\mathcal{M}}^h L^T = I$ for all $h \in \mathcal{M}_1 \subseteq \mathcal{M}$, then there exists a code (A, G, B) covering failure of links in $\mathcal{M}_1$, such that $A_{\mathcal{J}} = Q$ and $G_{\mathcal{J}\times\mathcal{J}} = J$. Conversely, if (A, G, B) is a code in which no link in $\mathcal{M}$ feeds into another, and (A, G, B) covers links in $\mathcal{M}_1 \subseteq \mathcal{M}$, then there exists some $r \times m$ matrix L such that $Q = A_{\mathcal{J}}$ and $J = G_{\mathcal{J}\times\mathcal{J}}$ satisfy $QJ_{\mathcal{M}}^h L^T = I$ for $h \in \mathcal{M}_1$.*

Proof: Extend (Q, J) to a valid interior code (A, G), where $A_{\mathcal{J}} = Q$ and $G_{\mathcal{J}\times\mathcal{J}} = J$, by having each link $k \in \mathcal{M}$ transmit its signal along the path P_k, such that the terminal link on P_k carries the same signal as link k. Then the receiver matrix B whose columns for terminal links on paths P_k are the same as the corresponding columns k of L, and zero for other terminal links, satisfies $AG^hB^T = QJ^h_{\mathcal{M}}L^T = I \;\forall\; h \in \mathcal{M}_1$.

For the converse, note that

$$\begin{aligned} AG^hB^T &= \sum_{j\in\mathcal{T}} \underline{c}_j^h \underline{b}_j^T \\ &= \sum_{j\in\mathcal{T}} \sum_{\substack{l\,\in\,\mathcal{M}\\ l\,\neq\,h}} \underline{c}_l G(l,j) \underline{b}_j^T \end{aligned}$$

So we can construct a matrix L which satisfies the required property as follows:

$$L^T = \begin{bmatrix} \sum_{j\in\mathcal{T}} G(l_1,j)\,\underline{b}_j^T \\ \hline \vdots \\ \hline \sum_{j\in\mathcal{T}} G(l_m,j)\,\underline{b}_j^T \end{bmatrix}$$

where $l_1, \ldots, l_m$ are the links of $\mathcal{M}$ in the order they appear in $J_{\mathcal{M}}$. ∎

Lemma 9 *For a single receiver with t terminal links, an upper bound on the number of receiver-based codes required for the no failure scenario and single terminal link failures is*

$$\max\left(\left\lceil \frac{t}{t-r} \right\rceil, r\right) = \begin{cases} r+1 & \textit{for}\, r = 1 \;\; \textit{or} \;\; t-1 \\ r & \textit{for}\, 2 \leq r \leq t-2 \end{cases}$$

Proof: For $r = 1$, $\left\lceil \frac{t}{t-r} \right\rceil = 2$. Just two codes are needed as only one of the links needs to be active in each code. For $t = r+1$, $\left\lceil \frac{t}{t-r} \right\rceil = r+1$. We can cover each of the $r+1$ terminal links by a separate code, so $r+1$ codes suffice. For $2 \leq r \leq t-2$, consider any valid static code (A, G). Let $\underline{v}_1, \ldots, \underline{v}_r$ be r columns of $AG_{\mathcal{T}}$ that form a basis, and $\underline{w}_1, \ldots, \underline{w}_{t-r}$ the remaining columns. Assuming that all single link failures are recoverable, and that there are at least $r+2$ nonzero columns, we can find a set

$(\underline{v}_i, \underline{w}_{i'}, \underline{v}_j, \underline{w}_{j'})$ such that $\{\underline{w}_{i'}\} \cup \{\underline{v}_x \mid x \neq i\}$ and $\{\underline{w}_{j'}\} \cup \{\underline{v}_x \mid x \neq j\}$ have full rank. Then the links corresponding to $\underline{v}_i$ and $\underline{w}_{j'}$ can be covered by one code, the links corresponding to $\underline{v}_j$, $\underline{w}_{i'}$ and $\{\underline{w}_k \mid k = 1, \ldots, t-r,\ k \neq i', j'\}$ by another code, and the links corresponding to $\{\underline{v}_k \mid k = 1, \ldots, r,\ k \neq i, j\}$ by a separate code each. ■

Lemma 10 *For any set of $n \geq 2$ codes with a common (A, G) covering failures from a set $\mathcal{T}_1 \subseteq \mathcal{T}$ of terminal links, there exists a set of n or fewer non-active codes that cover failures in set $\mathcal{T}_1$.*

Proof: A set of two or more terminal links covered by a single active code carry signal vectors which are multiples of each other. One of the links can be arbitrarily designated as the primary link for the code, and the others the secondary links for the code. If all n codes are active codes which cover two or more terminal link failures, then only two ($\leq n$) non-active codes are required, one non-active in the primary links and the other non-active in the rest. Otherwise, there is some non-active code in the set, or some active code covering only one terminal link failure which can be replaced by a corresponding non-active code covering that link. The links covered by this non-active code can be covered together with the primary links of the active codes, with a single non-active code. The secondary links of the active codes can be covered by a separate non-active code. This forms a set of at most n non-active codes covering the same terminal link failures as the original set. ■

Corollary 4 *For receiver-based recovery, the minimum number of codes for terminal link failures can be achieved with non-active codes.*

Lemma 11 *Bounds on the number of receiver-based codes needed to cover the no-failure scenario and failures of links in $\mathcal{M}$, assuming they are recoverable, are given in the following table. These bounds are the same in the case where only non-active codes are used.*

lower bound	*upper bound*
$\left\lceil \frac{m}{m-r} \right\rceil$	$\max\left(\left\lceil \frac{m}{m-r} \right\rceil, r\right) = \begin{cases} r+1 & \text{for} \quad r = 1 \ \text{or} \ m-1 \\ r & \text{for} \quad 2 \leq r \leq m-2 \end{cases}$

Proof: It follows from Lemma 7 that if failure of some link in $\mathcal{J}$ is recoverable, it is recoverable for the related problem Π'. Any code (Q', J') covering failure of terminal links $h \in \mathcal{M}_1$ in problem Π' can be extended to obtain a code (A, G, B) covering links $h \in \mathcal{M}_1$ in the original problem (Lemma 8). We can thus apply the upper bound from Lemma 9 with m in place of t.

For the lower bound, from Lemma 6, a single non-active code in a valid receiver-based scheme can cover at most $m - r$ of the links in $\mathcal{M}$. By Corollary 4, restricting consideration to non-active codes does not increase the receiver-based lower bound for the related terminal link problem Π', which is also $\lceil \frac{m}{m-r} \rceil$, and so does not increase the receiver-based lower bound here. ■

Lemma 12 *A lower bound on the number of network-wide codes needed to cover the no-failure scenario and failures of links in $\mathcal{M}$, assuming they are recoverable, is given by $\lceil \frac{m+1}{m-r+1} \rceil$.*

Proof: It follows from Lemma 6 that a single non-active code covers the no-failure scenario and at most $m - r$ single link failures among links in $\mathcal{M}$, while a single active code covers at most $m - r + 1$ links in $\mathcal{M}$. Each code therefore covers at most $m - r + 1$ out of $m + 1$ scenarios of no failures and failures of links in $\mathcal{M}$. ■

Lemma 13 *For a single receiver, there exists a valid static interior code (A, G) such that no link feeds into more than one link in $\mathcal{M}$.*

Proof: From Corollary 3, assuming single link failures are recoverable in the original problem Π, single link failures are recoverable in the related problem Π'. Thus, a static interior code (Q, J) covering these failures exists for Π' [38]. This can be extended to a static interior code (A, G) in which no link in $\mathcal{M}$ feeds into another.

For any such code (A, G), suppose there is some link h which feeds into more than one link in $\mathcal{M}$. Let $\mathcal{M}^h = \{h_1, \ldots, h_x\}$ be the set of links in $\mathcal{M}$ that h feeds into, and let $\overline{\mathcal{M}^h} = \mathcal{M} - \mathcal{M}^h$. We will show that we can obtain from (A, G) a valid static code in which h feeds into only one link in $\mathcal{M}$.

Case 1: h feeds into some link h_i in $\mathcal{M}$ via some path P (which includes h and h_i) such that the code for each link $l \in P$ other than h is $Y(l) = f_{l',l}Y(l')$, where l' is the incident upstream link in P of l, and $f_{l',l}$ is a nonzero coefficient, i.e. the signal vector of each link in P is a multiple of the signal vector of h.

Consider a code (Q, D) on the related problem Π' defined earlier, where $Q = A_{\mathcal{J}}$ and

$$D(l_1, l_2) = \begin{cases} 0 & \text{for } l_1 \in P, l_2 \notin P \\ f_{l_1,l_2} & \text{otherwise,} \end{cases}$$

i.e. each link in P feeds only into its incident downstream link in P. Let $J = (I - D)^{-1}$.

Consider any link $h' \in P$. Note that $QJ^{h'}_{\mathcal{M}} = AG^{h}_{\mathcal{M}}$, which has full rank. For failure of any link $k \notin P$, $\underline{c}_h$ is available on h_i via P, so rank$(QJ^{k}_{\mathcal{M}})$ = rank$(AG^{k}_{\mathcal{M}})$=r. Thus, (Q, J) is a valid static code for failures in Π'.

The extension of code (Q, J) is then a valid static code for the original problem Π in which h feeds into only one link in $\mathcal{M}$.

Case 2: Coding occurs between h and each $h_i \in \mathcal{M}^h$, i.e. the signal vector for each h_i is a combination of the signal vector for h and some other signal vector, which we denote by $\underline{s}_i$. The signal vector for h_i, $i = 1, \ldots, x$, is then $\underline{s}_i + G(h, h_i)\underline{c}_h$.

We first show that there exists a proper subset $\mathcal{L} \subset \mathcal{M}$ such that $AG^{h}_{\mathcal{L}}$ has full rank and which does not include all links in $\mathcal{M}^h$, i.e. $\mathcal{M}^h \cap \overline{\mathcal{L}}$ is nonempty. Suppose that such a subset does not exist. Since $AG^{h}_{\mathcal{M}}$ has full rank and $m > r$, $AG^{h}_{\mathcal{M}}$ must have at least one proper subset of r independent columns. By supposition, any such subset contains $\{h_1, \ldots, h_x\}$, which requires $\{\underline{s}_1, \ldots, \underline{s}_x\}$ to be independent, and $\underline{s}_i$ to be out of the column space of $AG_{\overline{\mathcal{M}^h}}$ $\forall\ i = 1, \ldots, x$ (where $\underline{s}_i$, defined in the previous paragraph, is the contribution to h_i from other links besides h). Then $AG_{\overline{\mathcal{M}^h}}$ has rank at most $r - x$, and failure of any $h_i, i = 1, \ldots, x$ would leave $AG^{h_i}_{\mathcal{M}}$ with less than full rank, contradicting the fact that (A, G) is valid for any single link failure. Thus, there exists a proper subset $\mathcal{L} \subset \mathcal{M}$ such that $AG^{h}_{\mathcal{L}}$ has full rank and $\mathcal{M}^h \cap \overline{\mathcal{L}}$ is nonempty. Let h_j be some link in $\mathcal{M}^h \cap \overline{\mathcal{L}}$.

For a particular code, let a link that feeds into more than one link in $\mathcal{M}$, and whose signal vector is a linear combination of $\underline{c}_h$ and some other nonzero signal vector, be said to satisfy condition 1. We consider two cases:

Case 2a: There exists a set R of links forming a single path from h to h_j, including h and h_j, such that none of the links $h' \in R$ satisfy condition 1.

Consider the family of codes (Q, D) on the related problem Π' defined earlier, satisfying $Q = A_{\mathcal{J}}$ and

$$D(l_1, l_2) = \begin{cases} 0 & \text{for } l_1 \in R, l_2 \notin R \\ d_{l_1,l_2} & \text{otherwise.} \end{cases}$$

Let $\mathcal{D}$ be the set of possible values for D, corresponding to different choices of values for variables d_{l_1,l_2}, in this family of codes. We will show that any single link failure in Π' can be covered by (Q, D) for some $D \in \mathcal{D}$. It will then follow that there exists a static choice of $D \in \mathcal{D}$ such that (Q, D) is valid for all single link failures in Π', since the product of the transfer matrix determinants for individual link failures is a nonzero polynomial in the variables d_{l_1,l_2}, which has a nonzero solution in a sufficiently large finite field [40].

Let D' be the element of $\mathcal{D}$ obtained by setting each variable d_{l_1,l_2} to f_{l_1,l_2}, and let $J' = (I - D')^{-1}$.

First consider failure of any link $h' \in R$. We have $QJ'^{h'}_{\mathcal{L}} = AG^{h}_{\mathcal{L}}$ by the assumption of this case. Hence, failure of h' is covered by (Q, J').

Next, consider some link $k \notin R$. If $AG^{\{h,k\}}_{\mathcal{M}}$ has full rank, then so does $QJ'^{\{h,k\}}_{\mathcal{M}}$. Then, the matrix $D'' \in \mathcal{D}$ obtained from D' by setting to zero each variable $d_{h,l_2}\ \forall\, l_2$ (i.e. having h not feed into any link) is such that (Q, D'') covers k.

If $AG^{\{h,k\}}_{\mathcal{M}}$ has less than full rank, then its rank is $r - 1$ since $AG^{h}_{\mathcal{M}}$, which has rank r, has only one possibly independent additional column. $\underline{c}_h$ is not in the column space of $AG^{\{h,k\}}_{\mathcal{M}}$, since otherwise $AG^{k}_{\mathcal{M}}$ would have rank no greater than $AG^{\{h,k\}}_{\mathcal{M}}$, contradicting the fact that $AG^{k}_{\mathcal{M}}$ has full rank. $\underline{s}^{k}_{j}$, the value on h_j after failure of h and k, is in the column space of $AG^{\{h,k\}}_{\mathcal{M}}$.

Now $AG^{\{h,k,h_j\}}_{\mathcal{M}}$ cannot have rank $\leq r-2$ since this would mean that $AG^{\{h,h_j\}}_{\mathcal{M}}$ has at most rank $r - 1$, and the full rank assumption on $AG^{h}_{\mathcal{L}}$, whose column space is contained in the column space of $AG^{\{h,h_j\}}_{\mathcal{M}}$, would be contradicted. Thus $AG^{\{h,k,h_j\}}_{\mathcal{M}}$

has rank $r-1$, which is the same as the rank of $AG_{\mathcal{M}}^{\{h,k\}}$. Since the column space of $AG_{\mathcal{M}}^{\{h,k,h_j\}}$ is contained in the column space of $AG_{\mathcal{M}}^{\{h,k\}}$, the column spaces must be equal. Hence $\underline{s}_j^k$ is in its column space while $\underline{c}_h$ is not, and thus $\underline{s}_j^k + G(h,h_j)\underline{c}_h$ is not in the column space. The column space of $QJ'^{k}_{\mathcal{M}}$ equals the column space of $AG_{\mathcal{M}}^{\{h,k,h_j\}} \cup \left(\underline{s}_j^k + G(h,h_j)\underline{c}_h\right)$, which has rank r.

Therefore, there exists some choice of values for variables in $\mathcal{D}$ such that (Q,D) is a valid static interior code for problem Π'. The extension of this static (Q,D) is a valid static code for the original problem Π in which link h feeds into only one link h_j in $\mathcal{M}$.

Case 2b: Every path from h to h_j contains some link that satisfies condition 1. Consider a set R' of links forming a path from h to h_j, and let $\tilde{h}$ be the furthest upstream link in R' that satisfies condition 1. We apply the same line of reasoning starting from the beginning of this proof, but with $\tilde{h}$ in place of h.

If case 1 or case 2a applies for (A,G) and $\tilde{h}$, then we can obtain a modified code (A',G') in which $\tilde{h}$ feeds into only one link in $\mathcal{M}$. Having eliminated one link from the set of those satisfying condition 1, we then re-apply the same reasoning from the beginning, this time for (A',G') and h.

If on the other hand case 2b applies for (A,G) and $\tilde{h}$, we proceed recursively, applying the same reasoning for (A,G) and a link downstream of $\tilde{h}$. If we come to a link $\hat{h}$ that is incident to a link in $\mathcal{M}$, then case 1 or case 2a will apply, allowing us to eliminate $\hat{h}$ from the set of links satisfying condition 1.

Throughout this procedure the number of links in $\mathcal{M}$ that h feeds into is monotonically decreasing, as is the number of its downstream links satisfying condition 1. Thus, the procedure terminates with a valid static interior code in which h feeds into only one link in $\mathcal{M}$. ■

Proof of Theorem 14a: We can find a valid static interior code (A,G) such that the subgraphs S_k of links which feed into each $k \in \mathcal{M}$ are link disjoint with each other, and the paths P_k along which k transmits to the receiver are also link disjoint (Lemma 13). A non-active code (A,G,B) which covers failure of link $k \in \mathcal{M}$ also covers failure of all links in the subgraph $S_k \cup P_k$, which we refer to as the *associated* *subgraph* of k. Thus, the bounds for receiver-based, or static, recovery here are the same as those in Lemma 11. An example of a valid static interior code achieving the lower bound with equality is an interior code (A, G) where $AG_{\mathcal{M}}$ is of the form shown in Figure 7-10.

$$
\begin{bmatrix}
x & 0 & \cdots & 0 & x & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & x & \cdots & 0 & x & 0 & \cdots & \cdots & & & & & & & & \vdots \\
\vdots & & \ddots & & \vdots & & & & & & & & & & & \vdots \\
\vdots & & & x & x & & & & & & & & & & & \vdots \\
\vdots & & & & & x & 0 & \cdots & 0 & x & & & & & & \vdots \\
\vdots & & & & & & x & \cdots & 0 & x & & & & & & \vdots \\
\vdots & & & & & & & \ddots & & \vdots & & & & & & \vdots \\
\vdots & & & & & & & & x & x & & & & & & \vdots \\
\vdots & & & & & & & & & & \ddots & & & & & \vdots \\
\vdots & & & & & & & & & & & x & 0 & \cdots & 0 & x \\
\vdots & & & & & & & & & & & & x & \cdots & 0 & x \\
\vdots & & & & & & & & & & & & & \ddots & & \vdots \\
0 & & & & & & & & & & & & & & x & x
\end{bmatrix}
\begin{matrix}
\vphantom{x} \\
\left.\vphantom{\begin{matrix}x\\x\\x\\x\end{matrix}}\right\} \left\lceil \frac{m}{m-r} \right\rceil - 1 \\
\left.\vphantom{\begin{matrix}x\\x\\x\end{matrix}}\right\} \left\lceil \frac{m}{m-r} \right\rceil - 1 \\
\vphantom{x} \\
\left.\vphantom{\begin{matrix}x\\x\\x\\x\end{matrix}}\right\} m - (m - r - 1) \left\lceil \frac{m}{m-r} \right\rceil - 1
\end{matrix}
$$

Figure 7-10: An example of an $AG_{\mathcal{M}}$ matrix in a receiver-based code that achieves the lower bound of $\left\lceil \frac{m}{m-r} \right\rceil$ codes, with $m - (m - r - 1) \left\lceil \frac{m}{m-r} \right\rceil \geq 2$.

For network-wide recovery, which includes receiver-based recovery as a special case, the maximum number of terminal link codes needed is no greater than that needed in receiver-based schemes.

For $r = m - 1 \geq 3$, we can obtain a bound tighter than the receiver-based bound. By Lemma 13, there exists a valid static interior code (A, G) such that no link feeds into more than one link in $\mathcal{M}$. Consider such a code (A, G). Define $\underline{e}_i \in \mathbb{F}_{2^u}^r$ to be the unit vector which has a 1 in the i^{th} position as its only nonzero entry. Since no link feeds into more than one link in $\mathcal{M}$, a column $AG_h, h \in \mathcal{M}$ can be set to $\underline{e}_i$, where process i is carried by link h, without affecting any of the other columns in $AG_{\mathcal{M}}$.

In order for single link failures to be recoverable, the $r \times (r + 1)$ submatrix $AG_{\mathcal{M}}$ of AG must have at least two nonzero entries in each row. It follows that we can find distinct $i, j, j' \in [1, r]$ and distinct $h, k \in \mathcal{M}$ such that $AG(i, h), AG(i, k)$ are nonzero, and either

(a) $AG(j, h), AG(j', k)$ are nonzero, or

(b) $AG(j, h), AG(j', h)$ are nonzero. In this case, we can switch labels among i, j, j' such that there exist paths in S_h from sources j and j' to h whose only common links are on a path from source i to h. Then paths from source i to the receiver through h and k can be covered by an active code, and the remaining links by $r-1$ non-active codes. This is because the remaining $r-1$ links in $\mathcal{M}$, and their associated subgraphs, can be covered by non-active codes corresponding to their receiver-based codes, and two paths from source j to h and from source j' to k (in case a) or to h (in case b), excluding the paths covered by the active code, can be covered with two of these non-active codes.

An example in which $r = m-1$, and r network-wide codes are needed is given in Figure 7-11. This is not the case for $r = 2 = m-1$, for which an example requiring 3 network-wide codes is given in Figure 7-12.

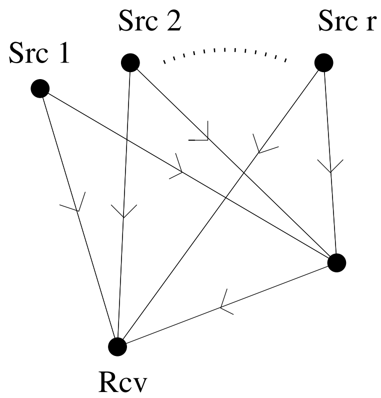


Figure 7-11: An example network in which $r = m-1$, which achieves the linear receiver-based upper bound of $r+1$ codes and the linear network-wide and nonlinear receiver-based upper bounds of r codes.

For $4 \leq r \leq m-2$, we can also obtain a bound tighter than the receiver-based bound. We consider two cases.

Case 1: There is a set of $r+2$ columns in $AG_{\mathcal{M}}$ which contains a basis and does not contain two pairwise dependent columns. We show that the set contains three pairs of columns such that each pair can be covered by a single non-active code, and that $r+2-3 = r-1$ non-active codes suffice to cover all columns.

Let the columns in this set be $\underline{u}_1, \ldots, \underline{u}_r, \underline{w}_1, \underline{w}_2$, where $\underline{u}_1, \ldots, \underline{u}_r$ form a basis, and let the remaining columns in $AG_{\mathcal{M}}$ be $\underline{w}_3, \ldots, \underline{w}_{m-r}$. Expressing each $\underline{w}_i$ as a linear combination $\underline{w}_i = \lambda_{i,1}\underline{u}_1 + \ldots + \lambda_{i,r}\underline{u}_r$, the pairwise independence of columns in

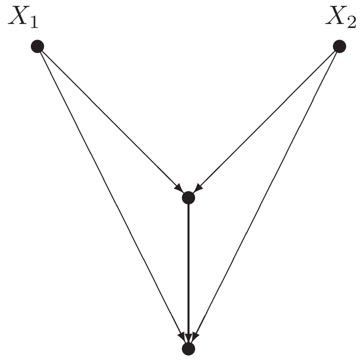


Figure 7-12: An example network in which $r = 2 = m - 1$, which achieves the linear network-wide upper bound of 3 codes.

the set implies that for $i = 1$ and $i = 2$, at least two of $\lambda_{i,1}, \ldots, \lambda_{i,r}$ are nonzero, and that there exist $k < l$ such that $\lambda_{1,k}\lambda_{2,l} \neq \lambda_{1,l}\lambda_{2,k}$. The last condition implies that $\lambda_{1,k}, \lambda_{2,l} \neq 0$ or $\lambda_{1,l}, \lambda_{2,k} \neq 0$; we assume wlog that $\lambda_{1,k}, \lambda_{2,l} \neq 0$. By the assumption of recoverability, at least one of $\lambda_{1,j}, \ldots \lambda_{m-r,j}$ is nonzero.

Case 1a: $\lambda_{1,k'}, \lambda_{2,l'} \neq 0$ for some k', l' such that k', l', k, l are all distinct. Then

$$\begin{aligned}
&\{\underline{u}_1, \ldots, \underline{u}_{l'-1}, \underline{u}_{l'+1}, \ldots, \underline{u}_r, \underline{w}_2\},\\
&\{\underline{u}_1, \ldots, \underline{u}_{k'-1}, \underline{u}_{k'+1}, \ldots, \underline{u}_r, \underline{w}_1\}, \text{ and}\\
&\{\underline{u}_1, \ldots, \underline{u}_{k-1}, \underline{u}_{k+1}, \ldots, \underline{u}_{l-1}, \underline{u}_{l+1}, \ldots, \underline{u}_r, \underline{w}_1, \underline{w}_2\}
\end{aligned}$$

are three full rank sets. Thus, links corresponding to each pair of columns $(\underline{w}_1, \underline{u}_{l'})$, $(\underline{w}_2, \underline{u}_{k'})$ and $(\underline{u}_k, \underline{u}_l)$ can be covered by one non-active code, along with links corresponding to any columns $\underline{w}_3, \ldots, \underline{w}_{m-r}$.

Case 1b: $\lambda_{1,k'}, \lambda_{2,k} \neq 0$ for some $k' \neq k, l$; and $\lambda_{2,j} = 0 \ \forall \ j \neq k, l$. Then $\lambda_{1,k'}\lambda_{2,l} \neq \lambda_{1,l}\lambda_{2,k'}$, so

$$\{\underline{u}_1, \ldots, \underline{u}_{k'-1}, \underline{u}_{k'+1}, \ldots, \underline{u}_{l-1}, \underline{u}_{l+1}, \ldots, \underline{u}_r, \underline{w}_1, \underline{w}_2\}$$

is a full rank set, as are

$$\{\underline{u}_1, \ldots, \underline{u}_{k-1}, \underline{u}_{k+1}, \ldots, \underline{u}_r, \underline{w}_2\} \text{ and}$$
$$\{\underline{u}_1, \ldots, \underline{u}_{l'-1}, \underline{u}_{l'+1}, \ldots, \underline{u}_r, \underline{w}_{m'}\}$$

where l' is distinct from k', k, l; and $m' \in \{1, 3, 4, \ldots, |\mathcal{M}| - r\}, \lambda_{m',l'} \neq 0$. Thus, links corresponding to the pair of columns $(\underline{u}_{k'}, \underline{u}_l)$ can be covered by a single code, along with links corresponding to any columns $\underline{w}_3, \ldots, \underline{w}_{m-r}$. The pairs $(\underline{w}_1, \underline{u}_k)$ and $(\underline{w}_2, \underline{u}_{l'})$ can each be covered by a single code.

Case 1c: $\lambda_{1,l}, \lambda_{2,l'} \neq 0$ for some $l' \neq k, l$; and $\lambda_{1,j} = 0 \ \forall \ j \neq k, l$. This case is similar to case 1b.

Case 1d: $\lambda_{1,l}, \lambda_{2,k} \neq 0$, $\lambda_{1,j} = 0, \lambda_{2,j} = 0 \ \forall \ j \neq k, l$. Links corresponding to columns $(\underline{u}_k, \underline{u}_l)$ can be covered by a single code along with links corresponding to any columns $\underline{w}_3, \ldots, \underline{w}_{m-r}$. Links corresponding to each pair of columns $(\underline{w}_1, \underline{u}_{l'})$ and $(\underline{w}_2, \underline{u}_{k'})$ can be covered by a single code, for some $k', l' \neq k, l$.

Case 2: For any basis set of r columns in $AG_{\mathcal{M}}$, there are no two columns among those remaining that are not multiples of each other or multiples of columns in the basis set.

Consider a pair of dependent columns. If each is a combination of two or more source processes, they can be set to different combinations of the same source processes while preserving the linear independence of any linearly independent subset of columns in $AG_{\mathcal{M}}$, in a sufficiently large finite field. This procedure can be repeatedly applied to remove pairwise dependence among columns involving two or more source processes, giving a new valid static code (A', G') in which any pair of dependent columns involves only one source process.

If (A', G') satisfies the condition of Case 1, then we know that $r - 1$ codes suffice. Otherwise, let us first consider the source processes and columns that are not part of pairwise dependent sets. Let $\tilde{r}$ be the total number of processes not involved in such sets, and $\tilde{\nu}$ be the number of columns that are not part of such sets. Note that $\tilde{r} \leq r - 1$ and $\tilde{r} \leq \tilde{\nu} - 1$.

By reasoning similar to our earlier analysis of receiver-based recovery, we have that the corresponding $\tilde{\nu}$ links and their associated subgraphs can be covered by $2 \leq \tilde{r}+1 \leq 3$ non-active codes if $\tilde{r} = 1, 2$, and by $\tilde{r}$ non-active codes if $2 \leq \tilde{r} \leq \tilde{\nu} - 2$. If $\tilde{r} = \tilde{\nu} - 1 \geq 3$, by reasoning similar to our analysis of network-wide recovery for $r = m - 1 \geq 3$, one active code and $\tilde{r} - 1 \geq 2$ non-active codes suffice to cover the $\tilde{\nu}$ links and their associated subgraphs. Any two non-active codes covering these links can also cover the remaining links corresponding to the dependent sets. Thus, $\tilde{r}+1 \leq 3$ codes suffice for $\tilde{r} \leq 2$, and $\tilde{r}$ codes suffice for $2 \leq \tilde{r} \leq \tilde{\nu}-2$ and $\tilde{r} = \tilde{\nu}-1 \geq 3$. In all these cases, the number of codes required is at most $r-1$, which is greater than or equal to 3.

For the remaining cases, the receiver-based upper bounds are also tight for the more general case of network-wide recovery.

The example network of Figure 7-13 achieves the receiver-based upper bound of r, and the network-wide upper bounds of r codes for $r = 3$, and $r - 1$ codes for $4 \leq r \leq m - 2$. ■

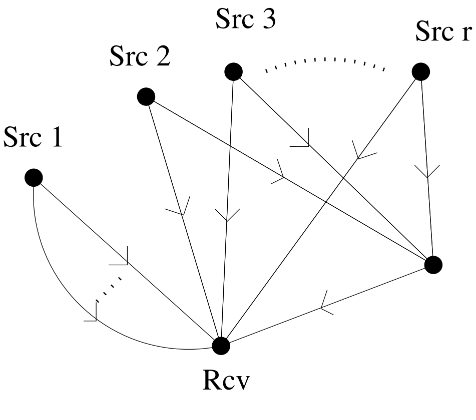


Figure 7-13: An example network which achieves the receiver-based upper bound of r, the network-wide upper bounds of r codes for $r = 3$, and $r-1$ codes for $4 \leq r \leq m-2$.

General case lower bound

Proof of Theorem 13:

Consider joining all receivers with $\max(m, 2r)$ links each to an additional node β'. If we consider β' to be the sole receiver node in the augmented network, the number of links in a minimum cut between the sources and this receiver is m, and there is a minimum cut of m links among the original links. The number of codes needed to cover links on this minimum cut is at least $\left\lceil \frac{m}{m-r} \right\rceil$ for receiver-based recovery and $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ for network wide recovery (Lemmas 11 and 12). Thus this represents a lower bound on the number of codes required to cover all links in the original problem.

An example which achieves the receiver-based lower bound with equality for any values of m and r is given in Figure 7-14, where the number of terminal links t_β of each receiver β is set to $2r_\beta$, twice the number r_β of processes needed by receiver β. Here, all links in $\mathcal{M}$ can be covered with $\left\lceil \frac{m}{m-r} \right\rceil$ non-active codes, two of which can cover at the same time all terminal links.

This example with $t_\beta = 2r_\beta$ for each receiver β also achieves the network-wide lower bound with equality when $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is not an integer. Let $\left\lceil \frac{m+1}{m-r+1} \right\rceil (m-r+1) = m+1+y$. Links in $\mathcal{M}$ can be covered with a set of $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ codes that includes $\min\left(\left\lceil \frac{m+1}{m-r+1} \right\rceil, y+1\right) \geq 2$ non-active codes, which can at the same time cover all the terminal links.

For the case where $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is an integer, however, covering links on the minimum cut with exactly $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ codes would allow for only one non-active code (Lemma 12), so this bound is not attained with equality for two or more receiver nodes. ∎

Upper bounds for all link failures, multicast case

Let m_β be the number of links in a minimum cut between the sources and a receiver β. From Lemmas 8 and 13, we know that for each receiver node β individually, there is a static solution for all single link failures in which each of m_β link-disjoint subgraphs feed into a different terminal link of β; each subgraph is a tree whose links are directed towards the root node β, with an unbranched portion between the root and the branches, which we term its *trunk*. We denote by $\mathcal{G}_x^i$, $i = 1, 2, \ldots, m_{\beta_x}$, the trees rooted at a receiver β_x. The trees corresponding to each receiver β_x can be partitioned into a number of forests such that failure of all links in any one forest

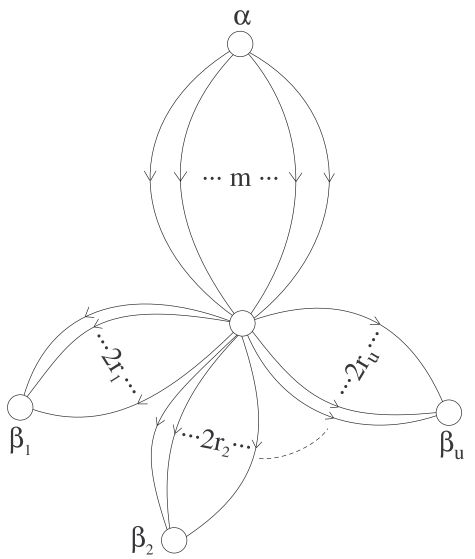


Figure 7-14: An example network which achieves the general case lower bounds of Theorem 2 with equality, where r_i is the number of processes received by receiver β_i.

leaves a subgraph of the network that satisfies the max-flow min-cut condition for receiver β_x. The number s_{β_x} of these forests is given by Theorem 14a.

Proof of Theorem 14b: We first analyze the two receiver case, considering three cases:

Case 1: $2 \leq r \leq m_{\beta_x} - 2$ for both receivers $\beta_x, x = 1, 2$. Then the trees $\mathcal{G}_x^i$, $i = 1, 2, \ldots, m_{\beta_x}$, associated with each receiver β_x, $x = 1, 2$, can be grouped into $s_{\beta_x} \leq r$ link-disjoint forests (Theorem 14a), such that failure of all links in any one forest leaves a subgraph of the network that satisfies the max-flow min-cut condition for receiver node β_x. Thus, at most r^2 codes are needed.

Case 2: $r = 1$. Consider the related problem where all but two terminal links of each receiver are deleted from the network such that the minimum cut between the source and each receiver is exactly two. This problem is also recoverable for all single link failures, and requires at least as many codes for failure recovery as the original problem. To see this, note that a valid code needs to use at least two paths, one from the source to each receiver. Thus, all links except for those on two paths, one from the source to each receiver, can be covered by a single code. Each link on these two paths must be covered by a code that uses an alternative pair of paths from the source to each receiver. Since the source-receiver paths in the related problem form a subset of those in the original problem, the related problem requires at least as many codes as the original problem.

Therefore, in finding an upper bound we can, without loss of generality, consider the case where the minimum cut capacity between the source and each receiver is exactly two. This puts us in case 3.

Case 3: one of the receivers, say β_1, has a minimum cut of $r+1$ links. We will show that there exists a set of paths sufficient for transmission to β_2, which does not intersect the trunk of some tree $\mathcal{G}_1^{\tilde{i}}$. Then the trunk of tree $\mathcal{G}_1^{\tilde{i}}$ can be covered by a single code. Its branches can be partitioned into sets $\mathcal{B}_1^k, k \leq r$, each paired with a distinct tree $\mathcal{G}_1^{\gamma_k}$, such that subtree of $\mathcal{G}_1^{\tilde{i}}$ excluding branches in set $\mathcal{B}_1^k$ can replace tree $\mathcal{G}_1^{\gamma_k}$ in a full rank set. Intersections between branches in set $\mathcal{B}_1^k$ and some tree $\mathcal{G}_2^j$ can then be covered together with intersections $(\mathcal{G}_1^{\gamma_k}, \mathcal{G}_2^j)$, if any.

If β_2 has a minimum cut of more than $r+1$ links, then $s_{\beta_i} \leq r$, and at most r^2+1 codes are required altogether.

If β_2 has a minimum cut of $r+1$ links, then by similar reasoning as for β_1, there exists some tree $\mathcal{G}_2^{\tilde{j}}$ whose trunk can be covered by a single code. Its branches can be partitioned into sets $\mathcal{B}_2^l, l \leq r$, each paired with a distinct tree $\mathcal{G}_2^{\gamma_l}$, such that subtree of $\mathcal{G}_2^{\tilde{j}}$ excluding branches in set $\mathcal{B}_2^l$ can replace tree $\mathcal{G}_2^{\gamma_l}$ in a full rank set. Then intersections between branches in set $\mathcal{B}_2^l$ and some tree $\mathcal{G}_1^i$ can be covered together with intersections $(\mathcal{G}_2^{\gamma_k}, \mathcal{G}_1^i)$, if any, and intersections between branches of $\mathcal{G}_2^{\tilde{j}}$ in set $\mathcal{B}_1^k$ and branches of $\mathcal{G}_2^{\tilde{j}}$ in set $\mathcal{B}_2^l$ can be covered together with intersections $(\mathcal{G}_1^{\gamma_k}, \mathcal{G}_2^{\gamma_l})$, if any.

Consider the following procedure that takes as inputs a set $\mathcal{T}$ of trees $\mathcal{G}_1^i$ and a set $\mathcal{P}$ of disjoint paths, and tries to shorten the paths to reduce the number of intersections with trees in $\mathcal{T}$. Let an intersection that is the furthest upstream on the trunk of some tree $\mathcal{G}_1^i$ be called a *leading* intersection. At each step, any path with a leading intersection that is not the furthest upstream intersection of the path is shortened by removing the portion of the path upstream of that leading intersection.

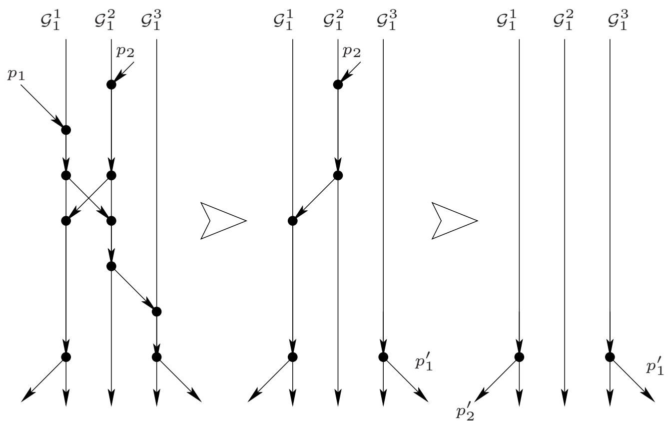


Figure 7-15: An illustration of the path shortening procedure. In the first step, path P_1 is shortened to form P_1' by removing the portion of P_1 upstream of its intersection with tree $\mathcal{G}_1^3$. In the second step, path P_2 is shortened to form P_2' by removing the portion of P_1 upstream of its intersection with tree $\mathcal{G}_1^1$.

The procedure ends when the leading intersection, if any, of each tree $\mathcal{G}_1^i$ is with the furthest upstream intersection of a path. An illustration of this procedure is given in Figure 7-15. We denote by $\mathcal{U} \subset \mathcal{T}$ the subset of trees with trunk intersections at the end of the procedure, and by $\mathcal{V} \subset \mathcal{P}$ the subset of paths with a leading intersection at the end of the procedure.

The sets $\mathcal{U}$ and $\mathcal{V}$ obtained at the end of the procedure are uniquely defined by the input sets, regardless of the choices made at steps where there is more than one candidate intersection that can be chosen by the modification procedure. First suppose to the contrary that two different sets $\mathcal{U}$ are obtained from the same inputs via two different sequences $\mathcal{S}_1$ and $\mathcal{S}_2$ of modifications. Then some tree $\mathcal{G}_1^i \in \mathcal{T}$ is in the set $\mathcal{U}$ for sequence $\mathcal{S}_1$ but not $\mathcal{S}_2$. This means that tree $\mathcal{G}_1^i$ has a leading intersection with some path P_j at the end of sequence $\mathcal{S}_1$, whereas tree $\mathcal{G}_1^i$ has no trunk intersections at the end of $\mathcal{S}_2$. Thus, $\mathcal{S}_2$ shortens path P_j such that its furthest upstream intersection is a leading intersection with some other tree $\mathcal{G}_1^{i'}$. The intersection $(\mathcal{G}_1^{i'}, P_j)$ is not however a leading intersection at the end of sequence $\mathcal{S}_1$; the leading intersection of tree $\mathcal{G}_1^{i'}$ is with some other path $P_{j'}$. This in turn means that $\mathcal{S}_2$ shortens path $P_{j'}$ such that its furthest upstream intersection is with yet another tree; continuing the argument in this fashion leads to a contradiction since the number of trees in $\mathcal{T}$ is finite.

Next suppose that two different sets $\mathcal{V}$ are obtained via two sequences $\mathcal{S}_1$ and $\mathcal{S}_2$ of modifications. Then some path P_j has a leading intersection at the end of one sequence $\mathcal{S}_1$ but not the other $\mathcal{S}_2$. This means that $\mathcal{S}_2$ does not modify P_j. The furthest upstream intersection of P_j at the end of $\mathcal{S}_1$ is with some tree $\mathcal{G}_1^i$; since this is not a leading intersection following $\mathcal{S}_2$, the leading intersection of tree $\mathcal{G}_1^i$ following $\mathcal{S}_2$ is with some other path $P_{j'}$. Path $P_{j'}$ is shortened by $\mathcal{S}_1$ such that its furthest upstream intersection is with some other tree $\mathcal{G}_1^{i'}$, whose leading intersection is with yet another path. Continuing similarly we reach a contradiction since the number of paths in $\mathcal{P}$ is finite.

This leads to the following property:

Property 1 *Let $\mathcal{P}'$ be the set of paths obtained from running the procedure on a set of paths $\mathcal{P}$ and a set of trees $\mathcal{T}'$. Running the procedure on $\mathcal{P}'$ and a set of trees $\mathcal{T}$ that is a superset of $\mathcal{T}'$ gives the same output sets $\mathcal{V}$ and $\mathcal{U}$ as running the procedure on $\mathcal{P}$ and $\mathcal{T}$.*

Thus, the output sets are unchanged if we carry out the procedure in two stages, first considering all intersections involving trees in a subset $\mathcal{T}' \subset \mathcal{T}$, then carrying out the procedure to completion on the entire set $\mathcal{T}$ of trees.

We will describe an algorithm for obtaining a set of paths that suffices for transmission to β_2 and has no intersections with the trunk of some tree $\mathcal{G}_1^{\tilde{i}}$. This algorithm involves one or more runs of the procedure described above. We denote by $\mathcal{T}_n, \mathcal{U}_n, \mathcal{V}_n$ respectively the sets $\mathcal{T}, \mathcal{U}, \mathcal{V}$ corresponding to the n^{th} run.

We set $\mathcal{T}_1$ to be the full set of trees $\mathcal{G}_1^i, i = 1, \ldots, r+1$, and $\mathcal{P}$ to be any set of r disjoint paths each joining a different source to β_2. If one of the trees in $\mathcal{T}_1$ has no intersections along its trunk, then we are done. Otherwise, consider the leading intersection of each tree and the furthest upstream intersection of each path. There exists a code in which the source vectors of the leading intersections of any r trees form a basis set. There exists also a code in which the source vectors of the furthest upstream intersection of each path form a basis set. Thus, there exists a code which satisfies both conditions simultaneously. We associate with each tree $\mathcal{G}_1^i$ the signal vector of its leading intersection in this code, and with each path the signal vector of its furthest upstream intersection in this code. We denote by $\mathcal{R}(\mathcal{Z})$ the set of source vectors of the trees or paths in a set $\mathcal{Z}$.

For the first run of the procedure, since there are $r+1$ trees in $\mathcal{T}_1$ and r paths in $\mathcal{P}$, the procedure ends with at least one tree whose trunk has no intersections.

Each run n of the procedure ends in one of the following two cases.

Case 3a: The set of paths at the end of the procedure suffice for transmission to β_2. Then we have a set of paths with the desired property.

Case 3b: The set of paths at the end of the procedure do not suffice for transmission to β_2. Then the set $\overline{\mathcal{V}}_n = \mathcal{P}_n - \mathcal{V}_n$ is non-empty, and some vector in the span of $\mathcal{R}(\overline{\mathcal{V}}_n)$ is also in the span of $\mathcal{R}(\mathcal{V}_n)$.

To see this, first note that that at the end of the procedure, every path in $\mathcal{V}_n$ forms the leading intersection of a distinct tree $\mathcal{G}_1^i$, and acquires the signal vector associated with that tree. Also, the signal vectors of any r trees form a basis set. If the redefined paths cannot carry a basis set, then at most $r-1$ trees $\mathcal{G}_1^i$ have leading intersections at the end of the procedure, and $|\mathcal{V}_n| \le r-1$. Next observe that since the vectors in $\mathcal{R}(\overline{\mathcal{V}}_n)$ are linearly independent, as are the vectors in $\mathcal{R}(\mathcal{V}_n) = \mathcal{R}(\mathcal{U}_n)$, any linearly dependent set of paths at the end of the procedure must include paths in both $\mathcal{V}_n$ and $\overline{\mathcal{V}}_n$.

Consider a basis set $\mathcal{W}_n$ for vectors that are both in the span of $\mathcal{R}(\mathcal{U}_n)$ as well as in the span of $\mathcal{R}(\overline{\mathcal{V}}_n)$. Each vector $\underline{t}_k \in \mathcal{W}_n, k = 1, \ldots, |\mathcal{W}_n|$, can be expressed as a linear combination of vectors forming a set $\mathcal{Y}_k \subseteq \mathcal{R}(\mathcal{U}_n)$, and paired with a vector $\underline{v}_k$ chosen from $\mathcal{Y}_k$ as follows. $\underline{t}_1$ is paired with an arbitrarily chosen vector $\underline{v}_1 \in \mathcal{Y}_1$. For subsequent vectors $\underline{t}_k, k \ge 2$, considered, if $\mathcal{Y}_k$ contains any vectors $\underline{v}_{k'}, k' < k$, Gaussian elimination is performed on vectors $\underline{t}_{k''}, k'' \ge k$, to obtain a vector in the span of a set $\mathcal{Y}'_k \subset \mathcal{R}(\mathcal{U}_n)$ that does not contain any vectors $\underline{v}_{k'}, k' < k$. This is possible because of the linear independence of vectors in $\mathcal{W}_n$. The vector $\underline{t}_k$ under consideration is then paired with an arbitrarily chosen vector $\underline{v}_k \in \mathcal{Y}'_k$. The pairings produced in this way have the property that the expression of any vector

$\underline{w} \in \operatorname{span}(\mathcal{W}_n)$ as a linear combination of vectors in $\mathcal{R}(\mathcal{U}_n)$ includes at least one vector $\underline{v}_k, 1 \leq k \leq |\mathcal{W}_n|$. The trees corresponding to vectors $\underline{v}_k$ are then removed from $\mathcal{T}_n$ to form set $\mathcal{T}_{n+1}$. The procedure is then run recursively on the new set of trees $\mathcal{T}_{n+1}$, which is a proper subset of the previous set $\mathcal{T}_n$.

Note that the set $\mathcal{V}_n$ formed by each run of the procedure is equal to or a subset of the sets $\mathcal{V}_{n'}$ formed by previous runs $n' < n$, and the set $\mathcal{T}_n - \mathcal{U}_n$ of each run is equal to or a subset of the sets $\mathcal{T}_{n'} - \mathcal{U}_{n'}$ from previous runs $n' < n$. This follows from property 1 and the following observations: that the set $\mathcal{T}$ of a run is a subset of that of previous runs, and that elements are added to but never removed from sets $\mathcal{V}$ and $\mathcal{T} - \mathcal{U}$ in the course of a procedure. This means that paths in the set $\overline{\mathcal{V}}_n$ of some run n will never have leading intersections in subsequent runs.

Next, we show that every run ends with a non-empty set $\mathcal{T} - \mathcal{U}$ of trees with no trunk intersections. As shown earlier, this is true for run $n = 1$. For $n > 1$, at most $|\overline{\mathcal{V}}_{n-1}|$ trees have been eliminated from $\mathcal{T}$ by the start of run n, so $|\mathcal{T}_n| \geq r+1-|\overline{\mathcal{V}}_{n-1}|$. Each run ends with each tree in $\mathcal{T}$ having either no trunk intersections, or having a leading intersection with the furthest upstream intersection of a path. At the end of run n, since at most $r - |\overline{\mathcal{V}}_{n-1}|$ paths can have leading intersections, at least one tree of $\mathcal{T}_n$ does not have a trunk intersection. Thus, $\mathcal{T}_n - \mathcal{U}_n$ is non-empty.

Finally, we show that any vector $\underline{w}$ in the span of $\mathcal{W}_n$ for some run n is independent of $\mathcal{R}(\mathcal{U}_j)$ for any subsequent run $j > n$. Consider the expression of $\underline{w}$ in terms of one or more vectors in the set $\mathcal{R}(\mathcal{U}_n)$. At least one of these vectors is not in the set $\mathcal{W}_j \subseteq \mathcal{U}_j$, its corresponding tree having been eliminated from $\mathcal{T}$ following run n. Now any vector can be expressed only as a linear combination of a subset of vectors in $\mathcal{R}(\mathcal{T}_1)$ or as a linear combination of the complementary subset of vectors in $\mathcal{R}(\mathcal{T}_1)$, otherwise there would exist a dependent set of r vectors in $\mathcal{R}(\mathcal{T}_1)$. Since the set $\mathcal{T}_j - \mathcal{U}_j$ is equal to or a subset of $\mathcal{T}_n - \mathcal{U}_n$, the set $\mathcal{T}_j - \mathcal{U}_j$ is disjoint with the set $\mathcal{U}_n$. The vectors in set $\mathcal{W}_j \subset \mathcal{R}(\mathcal{U}_j)$ are thus linearly independent with $\underline{w}$. As a result, the vectors in the set $\mathcal{W}$ corresponding to a run are independent of those in previous runs.

Since the total number of vectors in sets $\mathcal{W}$ is upper-bounded by $|\overline{\mathcal{V}}| \leq r$, and the set $\mathcal{W}$ for each run of the procedure ending in case 3b must be non-empty, the procedure eventually ends in case 3a.

This proves the result for the two receiver case.

For $d > 2$, the trees $\mathcal{G}_x^i$, $i = 1, 2, \ldots, m_{\beta_x}$, associated with each receiver β_x, $3 \leq x \leq d$, can be grouped into $s_{\beta_x} \leq r+1$ link-disjoint forests (Theorem 14a), such that failure of all links in any one forest leaves a subgraph of the network that satisfies the max-flow min-cut condition for receiver node β_x. Thus a set of links intersecting 0 or 1 of the forests associated with each receiver can be covered together.

Our analysis for the two receiver case partitions the links upstream of two receivers β_1 and β_2 into at most $r^2 + 2$ sets such that failure of all links in any one set leaves a subgraph of the network that satisfies the max-flow min-cut condition for receivers β_1 and β_2. Each of these partitions may contain links that are part of up to $r+1$ forests corresponding to receiver β_3, which have to be covered separately. Each of the resulting $\leq (r^2 + 2)(r + 1)$ subsets may in turn contain links that are part of $\leq r + 1$ such sets for receiver β_4, and so on. Thus, at most $(r^2 + 2)(r + 1)^{d-2}$ codes are required for d receivers. ■

We are not yet certain as to how tight the bounds are for the multi-receiver all link failures case. For the two-receiver case, an example in which $(r + 1)(r + 2)/2$ codes are needed is given in Figure 7-16. In this figure, there are $r+1$ paths leading to each receiver, which intersect each other in a stair-like pattern: the first path to Receiver 1 intersects one path to Receiver 2, the second path to Receiver 1 intersects two paths to Receiver 2, the third intersects three and so on. Each of the $(r + 1)(r + 2)/2$ intersections must be covered by a separate code.

The non-multicast case differs from the multicast case in that processes which are needed by one node but not another can interfere with the latter node's ability to decode the processes it needs. As a result, a static interior solution does not always exist, and the network management requirement for terminal link failures may exceed the corresponding upper bound from the multicast case. Unlike the multicast case where the number of codes for terminal link failures is bounded by $r + 1$, in the non-multicast case, the number of codes for terminal link failures can grow linearly in the

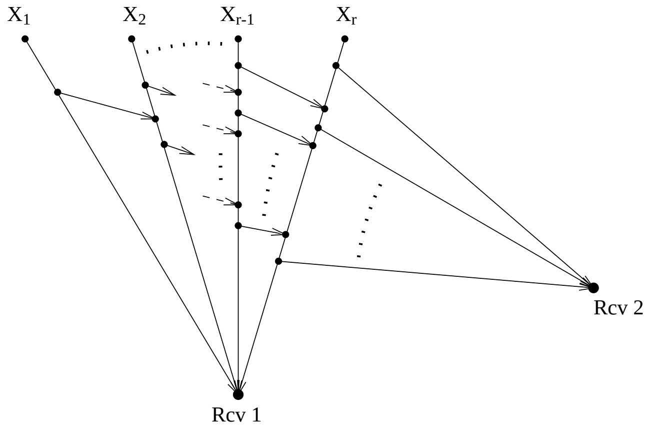


Figure 7-16: An example multicast problem in which $(r+1)(r+2)/2$ codes are needed for all link failures.

number of receivers.

Proof of Theorem 14c: We will use non-active codes in this proof. Let a set $\mathcal{S}$ of terminal links of a receiver β be called a *decoding set* for β in a given interior code if β can decode the processes it needs from links in $\mathcal{S}$, but not from any subset of $\mathcal{S}$. $\mathcal{S}$ is called a decoding set for β in a given failure scenario if $\mathcal{S}$ is a decoding set for β in some valid interior code under this scenario.

Note that $r \geq 2, d \geq 2$ for a non-multicast problem. From Theorem 12, at least 2 codes are required to cover failures of a receiver's terminal links. Consider a receiver β that has $\geq r+1$ terminal links, and any recoverable set of failures of one or more terminal links of other receivers. In any interior code (A, G) that is valid under failure of these terminal links, and in which all terminal links of β have nonzero signal vectors, either β has a decoding set of $\leq r-1$ links, or it has at least two possible choices of decoding sets of r links. All terminal links of β except those in a decoding set can be covered by (A, G). If β has a decoding set of r links, at least one of these can be covered by any interior code (A', G') valid under failure of another set of terminal links, and in which all terminal links of β have nonzero signal vectors. So at most $r-1$ of its terminal links require an additional code. ■

We have not yet determined whether this bound is tight. Figure 7-17 gives an example which comes close to this bound, requiring $\sum_{t_\beta \leq r}(t_\beta - 2) + \sum_{t_\beta \geq r+1}(r-1)$

codes. Here, each adjacent pair of receivers i and $i+1$ shares a common ancestral link $h_{i,i+1}$ which can carry two processes, each of which is needed by only one of the two receivers. Failure of any link to the left of j_i, other than $j_{i'}$, $i' < i$ requires $h_{1,2}$ to carry one of the processes only, and failure of any link to the right of k_{i+1}, other than $k_{i'}$, $i' > i+1$, requires $h_{1,2}$ to carry the other process only, necessitating separate codes.

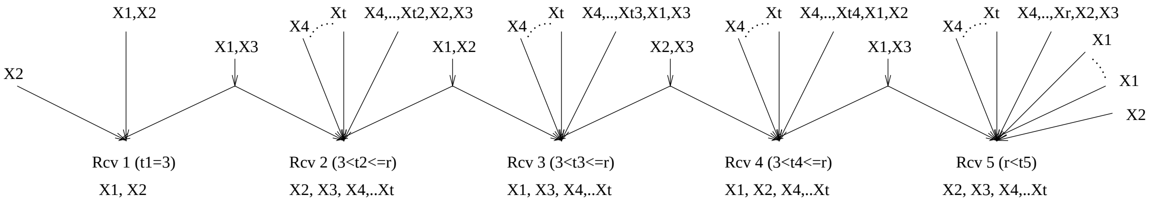


Figure 7-17: An example network in which $\sum_{t_\beta \leq r}(t_\beta - 2) + \sum_{t_\beta \geq r+1}(r-1)$ codes are needed.

7.4.4 Nonlinear receiver-based recovery

Proof of Theorem 15: We can view the signals on a receiver's terminal links as a codeword from a linear (t_β, r) code with generator matrix AG_β. The minimum number of nonlinear receiver codes required is the maximum number of codewords that can be the source of any one received codeword under different scenarios.

Assuming that zero signals are observed on failed links, no network management is needed for single link failures if each codeword differs from any other in at least 2 positions which are both nonzero in at least one of the codewords.

For a single receiver β, recovery from single terminal link failures with no network management requires the code with generator matrix AG_β to have minimum weight 2 and satisfy the property that for any pair of codewords which differ in only 2 places, one of them must have nonzero values in both places. Now if there were a code of weight 2, rank r and length $t = r+1$, it would be a maximum distance separable code, which has the property that the codewords run through all possible r-tuples in every set of r coordinates. In a set of r coordinates, where each entry is an element in $\mathbb{F}_q$, consider the $(q-1)r$ codewords with exactly 1 nonzero entry in this set of coordinates. For a weight 2 code, these $(q-1)r$ codewords must all be nonzero in the remaining coordinate. They must also all differ from each other in the remaining coordinate if they are to satisfy the property that for any pair of codewords which differ in only 2 places, one of them must have nonzero values in both places. This is possible for $r = 1$, but not for $r > 1$, as there are only $q - 1$ possible values for the remaining coordinate. There will be at least r different codewords which give the same received codeword for different failures. For $t \geq r+2$, there exist codes of weight 3 in some large enough finite field $\mathbb{F}_q$. A simple example is a network consisting of t parallel links between a single source of r processes and a receiver.

The linear receiver-based upper bounds of Lemma 9 apply since linear coding is a special case. For $2 \leq r \leq t-2$, the bound of r codes is tight, as shown in the example of Figure 7-18. For $r = 1$, there are at least two terminal links that carry the single process, and loss of either link leaves the receiver able to decode using an OR operation, so one code suffices. For $r = t - 1$, suppose we need $r + 1$ codes for each of the $r + 1$ terminal link failures. This means that there are $r + 1$ different combinations of source processes that give the same received codeword, each under a different terminal link failure, since no two combinations of source processes give the same received codeword under the same scenario. The common codeword would then have 0 in all $r + 1$ places, which implies that the weight of the code is 1. However, this is not possible in a valid static code as loss of a single link could then render two codewords indistinguishable. Thus at most r different codewords can be the same under different single link failures. An example in which $r = t - 1$, and r nonlinear receiver-based codes are needed is given in Figure 7-11.

Next we consider the multiple receiver case. We refer to the code generated by AG_β as a β code, and the codewords as β codewords. A β codeword under a single link failure of a receiver β cannot coincide with a different β codeword under no failures of terminal links of β, since this would imply that the β code has minimum distance 1, which would not be the case in a valid static code. So a receiver which receives a no-failure codeword can ignore management information regarding failures. Thus the management information does not need to distinguish among terminal link failures of different receivers. As such, a static code in a multiple receiver problem such that each receiver requires n_β nonlinear codes requires $\max_\beta n_\beta$ codes in total. ■

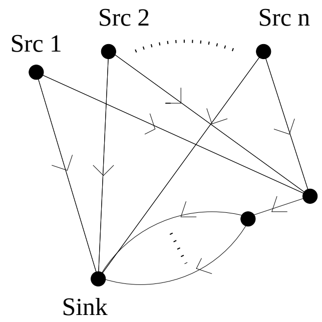


Figure 7-18: An example network in which $2 \leq r \leq t-2$, which achieves the nonlinear receiver-based upper bound of r codes.

7.4.5 Node-based management requirement

To prove Theorem 16, we first establish the following lemmas.

Lemma 14 *In a given network, for any set of non-active codes*

$$\{(A_1, G_1, B_1), (A_2, G_2, B_2), \ldots, (A_n, G_n, B_n)\}$$

there exists a set of receiver-based codes $\{(A, G, B_1'), (A, G, B_2'), \ldots, (A, G, B_n')\}$*, such that* (A, G, B_i') *covers the same terminal link failures as* (A_i, G_i, B_i)*, for all* $i = 1, \ldots, n$*.*

Proof: Each non-active code covers a set of terminal links $\mathcal{H}_i$ whose complement $\overline{\mathcal{H}}_i$ corresponds to columns of $AG_\mathcal{T}$ that contain a set of r independent columns. Let the nonzero entries of A and F be parameterized by elements forming a vector $\underline{\xi}$. There are submatrices $AG_\mathcal{T}^{\mathcal{H}_i}(\underline{\xi})$ consisting of r of these columns that have nonzero determinant $g_\mathcal{T}^{\mathcal{H}_i}(\underline{\xi})$. For any set of such codes, there exist static coefficients $\underline{\xi}$ in a large enough finite field such that all $g_\mathcal{T}^{\mathcal{H}_i}(\underline{\xi})$ are nonzero. ■

Corollary 5 *The terminal link failures covered by each code in a network-wide scheme can be covered by one or two codes in a receiver-based scheme.*

Proof: Terminal link failures covered by a single network-wide code active in those links correspond to columns in $AG_{\mathcal{T}}$ which are multiples of each other (Lemma 6). Only one of these columns is needed to form a basis, so a single non-active code can cover all but one of these links, and another non-active code can cover the remaining link. The result follows from applying Lemma 14. ■

Lemma 15 *If the no failure scenario and all single terminal link failures are covered by a set of n codes $\{(A_1, G_1, B), (A_2, G_2, B), \ldots, (A_n, G_n, B)\}$ having a common B matrix, then they can be covered by a set of n codes $\{(A, G, B_1), (A, G, B_2), \ldots, (A, G, B_n)\}$ with a common AG matrix.*

Proof: Since an active code cannot cover the no-failure scenario (Lemma 6), there is at least one non-active code. If codes $\{(A_1, G_1, B), (A_2, G_2, B), \ldots, (A_n, G_n, B)\}$ are all non-active, there is a set of n codes with common (A, G) that cover the same terminal link failures (Lemma 14).

Otherwise, there is at least one active code among them. We denote the set of terminal links covered by a code (A_i, G_i, B) by $\mathcal{H}_i$, and the set of remaining terminal links by $\overline{\mathcal{H}_i}$. Consider any active code (A_j, G_j, B) and any non-active code (A_k, G_k, B). Columns $\underline{b}_i, i \in \mathcal{H}_j$ are multiples of each other, i.e. $\underline{b}_i = \lambda_i \underline{v}$ for constants λ_i and a vector $\underline{v}$. Now

$$\sum_{i \in \overline{\mathcal{H}_k}} \underline{c}_{ki} \underline{b}_i = \sum_{i \in \overline{\mathcal{H}_k} \cap \overline{\mathcal{H}_j}} \underline{c}_{ki} \underline{b}_i + \left(\sum_{i \in \overline{\mathcal{H}_k} \cap \mathcal{H}_j} \lambda_i \underline{c}_{ki} \right) \underline{v}$$

has full rank. If $\{\underline{c}_{ki} | i \in \overline{\mathcal{H}_j} \cap \overline{\mathcal{H}_k}\}$ does not contain a full basis, then one of the columns $\underline{c}_{kh}, h \in \mathcal{H}_j$ is not in the range of $\{\underline{c}_{ki} | i \in \overline{\mathcal{H}_j} \cap \overline{\mathcal{H}_k}\}$. Then $\mathcal{H}' = \overline{\mathcal{H}_k} \cup \{h\}$ contains a full basis, i.e. $A_k G_{k\mathcal{H}'}$ has full rank. If $\{\underline{c}_{ki} | i \in \overline{\mathcal{H}_j} \cap \overline{\mathcal{H}_k}\}$ contains a full basis, h can be any link in $\mathcal{H}_j$. Thus, (A_k, G_k) is part of a valid non-active code (A_k, G_k, B_k) covering the rest of the links in $\mathcal{H}_j$ apart from h, together with links in $\mathcal{H}_k$.

Proceeding similarly, the secondary links of each active code can be covered together with some non-active code, and its primary link can be covered by a new non-active code. A set of n non-active codes covering the same failures as the original set can thus be constructed. By Lemma 14, there exists a set of n receiver-based codes covering the same failures. ■

Proof of Theorem 16: If interior nodes $i = 1, \ldots, x$ each switch among m_i codes respectively and the receiver switches among n codes, the node-based management requirement is $\sum_{i=1}^{x} \log_2 m_i + \log_2 n = \log_2 (\Pi_{i=1}^{x} m_i) n \geq \log_2 mn$, where m is the number of different values for AG among all the codes. $m \geq \Pi_{i=1}^{x} m_i$ because between two distinct values of AG, there is at least one interior node which switches code.

Let a set of codes covering the no-failure scenario and all terminal link failures be called *complete*. We show that for any complete set of network-wide codes with m values for AG and n values for B, there exists a complete set of $\leq mn$ receiver-based codes. Then the receiver-based management requirement is $\leq \log_2 mn$, which is less than or equal to the network-wide requirement.

Case 1: $m = 1$. There exists a complete set of $n = mn$ codes with a static AG matrix, which are receiver-based codes.

Case 2: $n = 1$. There exists a complete set of m codes with a static B matrix. By Lemma 15, there exists a complete set of $m = mn$ receiver-based codes with a static B matrix.

Case 3: $m \geq 2, n \geq 2$. If any set of $n_1 \geq 2$ codes $\{(A, G, B_1), (A, G, B_2), \ldots, (A, G, B_{n_1})\}$ has a common AG matrix, there is a corresponding set of $\leq n_1$ non-active codes covering the same terminal links (Lemma 10). Each of the remaining codes can be covered by one or two non-active codes (Corollary 5). Replacing active codes by non-active codes in this way, the maximum resulting number of non-active codes is mn. This is because each of the original codes is a pairing between one of m AG matrices and one of n B matrices. If there are codes corresponding to all mn combinations, then each code has a AG matrix that is the same as for $n-1$ other codes, and mn non-active codes suffice. If there are $k \geq 1$ AG matrices that are not common across two or more codes, then the number of non-active codes needed is at most

$(m-k)n + 2k = mn - k(n-2) \leq mn$ for $n \geq 2$. Thus, there exists a complete set of $\leq mn$ receiver-based codes (Lemma 14). ■

Chapter 8

Summary and future work

8.1 Summary

This thesis is an exploration of theoretical and operational networking issues from new perspectives that consider coding at network nodes.

We have presented a distributed randomized network coding approach which asymptotically achieves optimal capacity in multi-source multicast networks. We have given a general bound on the success probability of such codes for arbitrary networks, showing that error probability decreases exponentially with code length. Our analysis uses connections we make between network coding and network flows/bipartite matching, which also lead to a new bound on required field size for centralized network coding. We have also given tighter bounds for more specific acyclic networks, which show how redundant network capacity and link reliability affect the performance of randomized network coding. Two examples of scenarios in which randomized network coding shows benefits over routing approaches have been presented. These examples suggest that the decentralized nature and robustness of randomized network coding can offer significant advantages in settings that hinder optimal centralized network control.

We have further shown how to exploit the distributed and randomized nature of this approach to inexpensively add Byzantine fault detection capability without the use of cryptographic functions. This is done by augmenting each packet with a number of hash bits, each a simple polynomial function of the data bits. The overhead represented by the ratio of hash bits to data bits can be traded off against the detection probability. The effectiveness of our approach depends only on the inability of a Byzantine attacker to insert modified packets designed using knowledge of all other packets received by other nodes.

Taking a source coding perspective, we have shown that distributed randomized network coding effectively compresses correlated sources within a network, approaching optimal capacity with the length of the codes. We provide error exponents that generalize corresponding results for linear Slepian-Wolf coding.

Lastly, we have given a theoretical framework for quantifying essential network management needed for failure recovery, in terms of the number of different network behaviors, or codes, required under different failure scenarios. We have considered two types of recovery schemes, receiver-based and network-wide, and two formulations for quantifying network management, a centralized formulation and a node-based formulation. For the centralized formulation, we have given bounds, many of which are tight, on management requirements for various network connection problems in terms of basic network parameters. Our results include a lower bound for arbitrary connections and an upper bound for multi-transmitter multicast connections, for linear receiver-based and network-wide recovery from all single link failures. For the node-based formulation, we have shown that the minimum node-based requirement for failures of links adjacent to a single receiver is achieved with receiver-based schemes.

As the complexity of networks and network applications increases, so does the need for new techniques for operating and managing networks. We have shown how network coding can serve in developing fundamental network characterizations, as well as in enabling powerful new approaches to operational network issues.

8.2 Further work

Further work includes extensions of distributed randomized network coding to non-uniform code distributions, possibly chosen adaptively or with some rudimentary coordination, to optimize different performance goals, such as decoding complexity. Another question concerns selective placement of randomized coding nodes in networks where not all nodes have coding capability. There are interesting problems in distributed resource optimization and scheduling algorithms for networks with concurrent multicast and unicast connections. We may also consider protocol issues for different communication scenarios, and compare specific coding and routing protocols over a range of performance metrics.

On the network management side, one area of further work is network management needs for network connection problems in which certain links are known to fail simultaneously. For instance, if we model a large link as several parallel links, the failure of a single link may entail the failure of all associated links. Such dependence may significantly lower our network management requirements. Other directions for further work include extending our results to networks with cycles and delay, studying the capacity required for transmission of network management signals, and considering network management for wireless networks with ergodically varying link states. We expect that similar approaches to the ones presented in this thesis may be useful.

Appendix A

Original proofs of Theorems 1 and 3

The original proofs of Theorems 1 and 3 are based on the following lemma, which is essentially a slightly more general form of Theorem 3.

Lemma 16 *Let A be an arbitrary $r \times \nu$ matrix and F an arbitrary upper triangular $\nu \times \nu$ matrix with zeros on the main diagonal. For $1 \le h' \le h \le \nu$, let $S_{h',h}$ be the set of all sets of integers $\{e_1, e_2, \ldots, e_k\}$ such that $h' = e_1 < e_2 < \ldots < e_k = h$. Let $\mathcal{H} = \{h_1, \ldots, h_r\}$, where $1 \le h_1 < \ldots < h_r \le \nu$. Then $|AG_{\mathcal{H}}| =$*

$$\sum_{\substack{\{(h'_1,\ldots,h'_r):\\ 1\le h'_j \le h_j,\\ h'_i \ne h'_j\ \forall\, i\ne j\}}} \left| \begin{matrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{matrix} \right| \sum_{\substack{\{(\mathcal{E}_1,\ldots,\mathcal{E}_r):\\ \mathcal{E}_j \in S_{h'_j,h_j},\\ \mathcal{E}_i \cap \mathcal{E}_j = \emptyset\\ \forall\, i \ne j\}}} \prod_{j=1}^{r} g(\mathcal{E}_j)$$

Proof: It follows from the definitions of transfer matrices A and $G = I + F + F^2 + \ldots$ that $\underline{c}_h$ can be computed recursively as follows:

$$\underline{c}_1 = \underline{a}_1 \tag{A.1}$$

$$\underline{c}_h = \sum_{i=1}^{h-1} \underline{c}_i f_{i,h} + \underline{a}_h, \quad h = 2, 3, \ldots, \nu \tag{A.2}$$

Using the expression

$$\underline{c}_h = \sum_{i=1}^{h} \underline{a}_i \sum_{\mathcal{E} \in S_{i,h}} g(\mathcal{E})$$

for each column of $AG_{\mathcal{H}}$ and expanding the determinant linearly in all columns, we obtain

$$\begin{aligned}
|AG_{\mathcal{H}}| &= \begin{vmatrix} | & & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_r} \\ | & & | \end{vmatrix} \\
&= \sum_{\substack{\{(h'_1,\ldots,h'_r) : \\ 1 \le h'_j \le h_j \\ h'_i \ne h'_j \; \forall\, i \ne j\}}} \begin{vmatrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{vmatrix} \prod_{i=1}^{r} \sum_{\mathcal{E} \in S_{h'_i, h_i}} g(\mathcal{E}) \\
&= \sum_{\substack{\{(h'_1,\ldots,h'_r) : \\ 1 \le h'_j \le h_j \\ h'_i \ne h'_j \; \forall\, i \ne j\}}} \begin{vmatrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{vmatrix} \sum_{\substack{\{(\mathcal{E}_1,\ldots,\mathcal{E}_r) : \\ \mathcal{E}_j \in S_{h'_j, h_j}\}}} \prod_{j=1}^{r} g(\mathcal{E}_j)
\end{aligned}$$

The above expansion does not take into account dependencies among the columns $\underline{c}_h$. We can obtain an equivalent expression with fewer terms by using the following alternative sequence of expansions which takes the dependencies into account. We start by expanding the determinant of $AG_{\mathcal{H}}$ linearly in the ${h_r}^{th}$ column using Equation A.2:

$$\begin{aligned}
|AG_{\mathcal{H}}| &= \begin{vmatrix} | & & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_r} \\ | & & | \end{vmatrix} \\
&= \sum_{\substack{\{i \;:\; 1 \le i < h_r, \\ i \ne h_1,\ldots,h_{r-1}\}}} \begin{vmatrix} | & & | & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_{r-1}} & \underline{c}_i \\ | & & | & | \end{vmatrix} f_{i,h_r}
\end{aligned}$$

$$+\begin{vmatrix} | & & | & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_{r-1}} & \underline{a}_{h_r} \\ | & & | & | \end{vmatrix}$$

and proceed recursively, expanding each determinant linearly in its column $\underline{c}_h$ whose index h is highest, using Equation A.2 for $h > 1$ and Equation A.1 for $h = 1$. At each expansion stage, the expression for $AG_{\mathcal{H}}$ is a linear combination of matrix determinants. Each nonzero determinant corresponds to a matrix composed of columns $\{\underline{a}_{k_1}, \ldots, \underline{a}_{k_s}, \underline{c}_{k_{s+1}}, \ldots, \underline{c}_{k_r}\}$ such that $k_i \neq k_j\ \forall\ i \neq j$, and $\min(k_1, \ldots, k_s) > \max(k_{s+1}, \ldots, k_r)$. Its coefficient in the linear combination is a product of terms $f_{i,h}$ such that $h > k_{s+1}, \ldots, k_r$, and is of the form $\prod_{j=1}^{r} g(\mathcal{E}_j)$ where $\mathcal{E}_j \in S_{h_{j'},h_j}$ and $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset\ \forall\ i \neq j$. We can show by induction that these properties hold for all nonzero determinant terms in the course of the expansion. The expansion terminates when the expression is a linear combination of determinants of the form $\left|\underline{a}_{l_1} \ldots \underline{a}_{l_r}\right|$, at which point we have the desired expression. ■

Proof of Theorem 3: The result follows from Lemma 16 by noting that each set $\mathcal{E} = \{e_1, e_2, \ldots, e_k\}$ such that $g(\mathcal{E}) \neq 0$ corresponds to a network path consisting of links $e_1, \ldots, e_k$; that the condition $\mathcal{E}_j \cap \mathcal{E}_k = \emptyset$ for all $j \neq k,\ 1 \leq j, k \leq r$ implies that the corresponding paths $\mathcal{E}_1, \ldots, \mathcal{E}_r$ are disjoint; and that $\left|\underline{a}_{h'_1} \ldots \underline{a}_{h'_r}\right|$ is nonzero only when links $h_{j'}$ are source links of r different sources and carry r independent signals. ■

Proof of Theorem 1: We prove that this result holds for any set of matrices (A, F, B)[1] where A and B are arbitrary $r \times \nu$ matrices, and F is an arbitrary upper triangular $\nu \times \nu$ matrix with zeros on the main diagonal.

$$|M_1| = \left| \begin{bmatrix} | & & | \\ \underline{c}_1 & \cdots & \underline{c}_\nu \\ | & & | \end{bmatrix} \begin{bmatrix} -\underline{b}_1^T- \\ \vdots \\ -\underline{b}_\nu^T- \end{bmatrix} \right|$$

[1]We drop the subscript β from B_β for notational simplicity.

$$= \begin{vmatrix} | & & | \\ \sum_{i=1}^{\nu} \underline{c}_i b_{1,i} & \ldots & \sum_{i=1}^{\nu} \underline{c}_i b_{r,i} \\ | & & | \end{vmatrix}$$

Expanding $|M_1|$ linearly in each column and using the determinant expansion

$$\det M = \sum_{\text{all permutations } p} (\operatorname{sign} p) M(p(1),1) \ldots M(p(r),r)$$

we have

$$\begin{aligned}
&|M_1| \\
&= \sum_{\substack{\{(h_1,\ldots,h_r) : \\ 1 \le h_1 < h_2 \\ \ldots < h_r \le \nu\}}} \sum_{\text{all permutations } p} \begin{vmatrix} | & & | \\ \underline{c}_{h_{p(1)}} b_{1,h_{p(1)}} & \ldots & \underline{c}_{h_{p(r)}} b_{r,h_{p(r)}} \\ | & & | \end{vmatrix} \\
&= \sum_{\substack{\{(h_1,\ldots,h_r) : \\ 1 \le h_1 < h_2 \\ \ldots < h_r \le \nu\}}} \sum_{\text{all permutations } p} (\operatorname{sign} p^{-1}) \\
&\qquad \begin{vmatrix} | & & | \\ \underline{c}_{h_1} b_{p^{-1}(1),h_1} & \ldots & \underline{c}_{h_r} b_{p^{-1}(r),h_r} \\ | & & | \end{vmatrix} \\
&= \sum_{\substack{\{(h_1,\ldots,h_r) : \\ 1 \le h_1 < h_2 \\ \ldots < h_r \le \nu\}}} \sum_{\text{all permutations } p} (\operatorname{sign} p^{-1}) \\
&\qquad \sum_{\text{all permutations } \tilde{p}} (\operatorname{sign} \tilde{p}) c_{\tilde{p}(1),h_1} b_{p^{-1}(1),h_1} \ldots c_{\tilde{p}(r),h_r} b_{p^{-1}(r),h_r} \\
&= \sum_{\substack{\{(h_1,\ldots,h_r) : \\ 1 \le h_1 < h_2 \\ \ldots < h_r \le \nu\}}} \sum_{\text{all permutations } \tilde{p}} (\operatorname{sign} \tilde{p}) c_{\tilde{p}(1),h_1} \ldots c_{\tilde{p}(r),h_r} \\
&\qquad \sum_{\text{all permutations } p} (\operatorname{sign} p^{-1}) b_{p^{-1}(1),h_1} \ldots b_{p^{-1}(r),h_r}
\end{aligned}$$

$$= \sum_{\substack{\{(h_1,\ldots,h_r): \\ 1\le h_1 < h_2 \\ \ldots < h_r \le \nu\}}} \begin{vmatrix} | & & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_r} \\ | & & | \end{vmatrix} \begin{vmatrix} - & \underline{b}_{h_1}^T & - \\ & \vdots & \\ - & \underline{b}_{h_r}^T & - \end{vmatrix} \tag{A.3}$$

By Lemma 16,

$$\begin{vmatrix} | & & | \\ \underline{c}_{h_1} & \cdots & \underline{c}_{h_r} \\ | & & | \end{vmatrix} = \sum_{\substack{\{h'_1,\ldots,h'_r: \\ 1\le h'_j \le h_j, \\ h'_i \ne h'_j \ \forall\ i\ne j\}}} \begin{vmatrix} | & & | \\ \underline{a}_{h'_1} & \cdots & \underline{a}_{h'_r} \\ | & & | \end{vmatrix}$$

$$\sum_{\substack{\{\mathcal{E}_1,\ldots,\mathcal{E}_r: \\ \mathcal{E}_j \in S_{h'_j,h_j}, \\ \mathcal{E}_j \cap \mathcal{E}_k = \emptyset \\ \forall\ j\ne k\}}} \prod_{j=1}^{r} g(\mathcal{E}_j) \tag{A.4}$$

First we show that there is a bijective correspondence between terms of $|M_1|$ and terms of $|M_2|$.

Each product term of $|M_1|$ is of the form

$$\prod_{i=1}^{r} a_{p(i),h'_i} f_{e^i_1,e^i_2} \cdots f_{e^i_{k_i-1},e^i_{k_i}} b_{p_b(i),h_i}$$

where p and p_b are permutations acting on $\{1,\ldots,r\}$, $\{e^i_1,\ldots,e^i_{k_i}\} \in S_{h'_i,h_i}$, and $\{e^i_1,\ldots,e^i_{k_i}\} \cap \{e^j_1,\ldots,e^j_{k_j}\} = \emptyset\ \forall\ i \ne j$. Let $\mathcal{C}_1$ be the set of variables

$$\{a_{p(i),h'_i}, f_{e^i_1,e^i_2}, \ldots, f_{e^i_{k_i-1},e^i_{k_i}}, b_{p_b(i),h_i} : i = 1,\ldots,r\}$$

in a given product term. The conditions on $\{h'_i, e^i_1, \ldots, e^i_{k_i}, h_i : i = 1,\ldots,r\}$ imply that no two variables in $\mathcal{C}_1$ appear in the same row or the same column in matrix M_2, and that if a column z in M_2 does not contain any of these variables, then $z \notin \bigcup_{i=1}^{r} \mathcal{E}_i$, so row $r+z$ also does not contain any of these variables. For such z, we append to $\mathcal{C}_1$ $M_2(r+z,z) = 1$. Then $\mathcal{C}_1$ comprises $r+\nu$ variables each occupying a different row and column of the $(r+\nu)\times(r+\nu)$ matrix M_2. The variables in $\mathcal{C}_1$ are thus variables of a product term in $|M_2|$.

Conversely, a product term of $|M_2|$ comprises $m+r$ entries of M_2, each from a different row and column of M_2. For any such nonzero product term, the variables form a set $\mathcal{C}_2 = \mathcal{A} \cup \mathcal{B} \cup \mathcal{F} \cup \mathcal{I}$, where

$$\begin{aligned}
\mathcal{A} &= \{a_{p_a(j),\hat{h}_j} : j = 1,\ldots,r;\ \hat{h}_1 < \ldots, \hat{h}_r\} \\
\mathcal{B} &= \{b_{p_b(i),h_i} : i = 1,\ldots,r;\ h_1 < \ldots < h_r\} \\
\mathcal{F} &= \{f_{x_l,y_l} : l = 1,\ldots,|\mathcal{F}|;\ y_1 < \ldots < y_{|\mathcal{F}|}\} \\
\mathcal{I} &= \{M_2(r+z_l, z_l) = 1 : l = 1,\ldots,|\mathcal{I}|\},
\end{aligned}$$

p_a and p_b are permutations acting on $\{1,\ldots,r\}$, and $\{h_1,\ldots,h_r,x_1,\ldots,x_{|\mathcal{F}|},z_1,\ldots,z_{|\mathcal{I}|}\}$ and $\{\hat{h}_1,\ldots,\hat{h}_r,y_1,\ldots,y_{|\mathcal{F}|},z_1,\ldots,z_{|\mathcal{I}|}\}$ are permutations of $\{1,\ldots,\nu\}$.

For $i = 1,\ldots,r$, h_i is equal to some y_l or some $\hat{h}_j$. Let $e_0^i = h_i$. If $e_0^i = y_l$, set $e_{-1}^i = x_l$; e_{-1}^i is in turn equal to some y_l or some $\hat{h}_j$. We proceed in this way, defining $e_{-2}^i, e_{-3}^i, \ldots$, until we reach an index e_{-k}^i that is equal to some $\hat{h}_j$. We then set $k_i = k+1$, $h_i' = \hat{h}_j$, and $e_{k_i-l}^i = e_{-l}^i\ \forall\ l = 0,\ldots,k_i-1$. With these definitions, $\mathcal{C}_2$ becomes $\{a_{p_a'(i),h_i'}, f_{e_1^i,e_2^i},\ldots,f_{e_{k_i-1}^i,e_{k_i}^i}, b_{p_b(i),h_i} : i = 1,\ldots,r\} \cup \big\{M_2(r+z,z) = 1 : z \notin \bigcup_{i=1}^r \{e_1^i,\ldots,e_{k_i}^i\}\big\}$, where the set of indices $\{h_i, h_i', e_1^i,\ldots,e_{k_i}^i : i = 1,\ldots,r\}$ so defined satisfies $h_i' = e_1^i < \ldots < e_{k_i}^i = h_i$ and $\{e_1^i,\ldots,e_{k_i}^i\} \cap \{e_1^j,\ldots,e_{k_j}^j\} = \emptyset\ \forall\ i \neq j$. From Equations A.3 and A.4, we can see that $\mathcal{C}_2$ comprises variables of some product term in $|M_1|$.

It remains to show equality of the sign of the product term in $|M_1|$ and that in $|M_2|$ whose variables constitute the same set $\mathcal{C} = \mathcal{A} \cup \mathcal{B} \cup \mathcal{F} \cup \mathcal{I}$, where

$$\begin{aligned}
\mathcal{A} &= \{a_{p(i),h_i'} : i = 1,\ldots,r\} \\
&= \{a_{p_a(j),\hat{h}_j} : j = 1,\ldots,r;\ \hat{h}_1 < \ldots, \hat{h}_r\} \\
\mathcal{B} &= \{b_{p_b(i),h_i} : i = 1,\ldots,r;\ h_1 < \ldots < h_r\} \\
\mathcal{F} &= \{f_{e_1^i,e_2^i},\ldots,f_{e_{k_i-1}^i,e_{k_i}^i} : i = 1,\ldots,r\} \\
&= \{f_{x_l,y_l} : l = 1,\ldots,|\mathcal{F}|;\ y_1 < \ldots < y_{|\mathcal{F}|}\}
\end{aligned}$$

$$\mathcal{I} \;=\; \{M_2(r+z_l, z_l) = 1 : l = 1, \ldots, |\mathcal{I}|\}$$

Let the corresponding product terms in $|M_1|$ and $|M_2|$ be $\sigma_1^{\mathcal{C}}\pi_{\mathcal{C}}$ and $\sigma_2^{\mathcal{C}}\pi_{\mathcal{C}}$ respectively, where $\pi_{\mathcal{C}} = \prod_{\xi\in\mathcal{C}} \xi$, and $\sigma_1^{\mathcal{C}}$, $\sigma_2^{\mathcal{C}}$ are the respective signs, equal to 1 or -1.

From Equations A.3 and A.4, we see that $\sigma_1^{\mathcal{C}} = (\operatorname{sign} p)(\operatorname{sign} p_b)$. Let p' be the permutation that sorts $\{h'_1, \ldots, h'_r\}$ in ascending order, i.e. $h'_{p'(1)} < \ldots < h'_{p'(r)}$. Then $p(p'(i)) = p_a(i)$. So $\operatorname{sign} p = (\operatorname{sign} p')(\operatorname{sign} p_a)$, and $\sigma_1^{\mathcal{C}} = (\operatorname{sign} p_a)(\operatorname{sign} p_b)(\operatorname{sign} p')$.

Consider the following procedure for determining $\operatorname{sign} p'$, consisting of a series of steps in which the variables $f_{x_l,y_l} \in \mathcal{F}$ are considered one by one in ascending order of index l. We maintain an ordered set $\mathcal{S} = \{s_1, \ldots, s_r\}$ of distinct elements from $[1,\nu]$, and an ordered set $\mathcal{Q}$ of the indices of elements of $\mathcal{S}$ sorted in ascending order, i.e. $\mathcal{Q} = \{q_1, \ldots, q_r\}$ where $s_{q_1} < \ldots < s_{q_r}$. Each s_i is initialized to h'_i. We carry out the following procedure for $l = 1, \ldots, |\mathcal{F}|$ in order. At each step l, $x_l = s_{\gamma_l}$ for some index $\gamma_l \in [1,r]$. We set $s_{\gamma_l} = y_l$, and let n_l be the number of indices i for which $x_l < s_i < y_l$. If $n_l \geq 1$, the change in $\mathcal{Q}$ is a cyclic permutation p_l of $n_l + 1$ elements. If $n_l = 0$, there is no change in $\mathcal{Q}$, and p_l is the identity permutation. We continue in this manner, noting that at every step, for the index y_l under consideration, all indices less than y_l are either equal to some s_i or some x_k where $y_k < y_l$. Since all the x_k's are distinct, x_l must equal some s_i.

At the end of the procedure, $s_i = h_i \; \forall\, i$, so the elements of $\mathcal{S}$ are in ascending order and $\mathcal{Q} = \{1, \ldots, r\}$. Permutation p' is equal to the composition of the cyclic permutations p_l, $l = 1, \ldots, |\mathcal{F}|$. Since $\operatorname{sign} p_l = (-1)^{n_l}$, $\operatorname{sign} p' = (-1)^{\sum_{l=1}^{|\mathcal{F}|} n_l}$.

Next we determine $\sigma_2^{\mathcal{C}}$. Let $M_{\mathcal{C}}, A_{\mathcal{C}}$ and $B_{\mathcal{C}}$ be the matrices obtained from M_2, A and B respectively by setting to 0 all entries involving variables not in $\mathcal{C}$, and let ξ_j be the nonzero entry in column j of $M_{\mathcal{C}}$. Let λ be the number of inversions in $M_{\mathcal{C}}$, where an inversion is a pair of nonzero entries at positions (q_1^i, q_1^j), (q_2^i, q_2^j) such that $q_1^i < q_2^i$ and $q_2^j < q_1^j$. Then $\sigma_2^{\mathcal{C}} = (-1)^{\lambda + |\mathcal{F}|}$, since each entry involving a variable in $|\mathcal{F}|$ has a negative sign in M_2.

For each j, let u_j be the number of inversions involving ξ_j and entries ξ_k, $k > j$. Then $\sigma_2^{\mathcal{C}} = (-1)^{\sum_{j=1}^{\nu+r} u_j + |\mathcal{F}|} = (-1)^{U_a + U_b + U_f + U_i + |\mathcal{F}|}$, where $U_a = \sum_{j:\xi_j\in\mathcal{A}} u_j$, $U_b =$

$\sum_{j:\xi_j\in\mathcal{B}} u_j$, $U_f = \sum_{j:\xi_j\in\mathcal{F}} u_j$ and $U_i = \sum_{j:\xi_j\in\mathcal{I}} u_j$.

If $\xi_j \in \mathcal{A}$ is involved in an inversion with ξ_k, $k > j$, then ξ_k must be a term in $\mathcal{A}$ as it is in a smaller-numbered row. Thus, the number of inversions involving entries in $\mathcal{A}$ is equal to the number of inversions in the $r \times r$ submatrix of $A_{\mathcal{C}}$ consisting of columns j for which $\xi_j \in \mathcal{A}$. So $(-1)^{U_a} = \text{sign } p_a$. Similarly, if $\xi_j \in \mathcal{B}$ is involved in an inversion with ξ_k, $k > j$, then ξ_k must be a term in $\mathcal{B}$ as it is in a larger-numbered column. So $(-1)^{U_b} = \text{sign } p_b$.

For j such that $\xi_j \in \mathcal{F}$, carry out the procedure described earlier that considers the terms $\xi_j = f_{x_l,y_l}$ one by one in ascending order of index l. At each step we compute $\mathcal{S}$, $\mathcal{Q}$ and n_l as before, noting that $x_l = s_{\gamma_l}$ for some $\gamma_l \in [1, r]$, and that the entries ξ_k, $k > y_l$, with which f_{x_l,y_l} is involved in inversions are $\{\xi_k : k = s_i > y_l\} \cup \{f_{x_g,y_g} : x_g = s_i < x_l, y_g = k > y_l\} \cup \{b_{k-\nu,g} : g = s_i < x_l\}$. These are in bijective correspondence with the elements of the set $\{s_i : s_i < x_l = s_{\gamma_j} \text{ or } s_i > y_l\}$. Thus, $u_j = r - 1 - n_l$.

For j such that $\xi_j \in \mathcal{I}$, there are exactly $j-1$ entries in columns $1, \ldots, j-1$, and exactly $\nu - j$ entries in rows $j+r+1, \ldots, \nu+r$ which are not involved in inversions with ξ_j. Thus, $u_j = \nu + r - 1 - (j - 1 + \nu - j) = r$.

Combining these expressions, and noting that $\mathcal{I} = \nu - r - |\mathcal{F}|$, we have

$$\begin{aligned}
U_f + U_i &= |\mathcal{F}|(r-1) - \sum_{l=1}^{|\mathcal{F}|} n_l + (\nu - r - |\mathcal{F}|)r \\
&= (\nu - r)r - |\mathcal{F}| - \sum_{l=1}^{|\mathcal{F}|} n_l \\
\sigma_2^{\mathcal{C}} &= (-1)^{U_a}(-1)^{U_b}(-1)^{U_f+U_i+|\mathcal{F}|} \\
&= (\text{sign } p_a)(\text{sign } p_b)(-1)^{(\nu-r)r-\sum_{l=1}^{|\mathcal{F}|} n_l}
\end{aligned}$$

If r is even, then $(\nu - r)r$ is even, and

$$\begin{aligned}
\sigma_2^{\mathcal{C}} &= (\text{sign } p_a)(\text{sign } p_b)(-1)^{-\sum_{l=1}^{|\mathcal{F}|} n_l} \\
&= (\text{sign } p_a)(\text{sign } p_b)(-1)^{\sum_{l=1}^{|\mathcal{F}|} n_l}
\end{aligned}$$

$$= \sigma_1^{\mathcal{C}}$$

If r is odd, then $(\nu - r)r$ is even if ν is odd, and odd if ν is even. So $\sigma_2^{\mathcal{C}} = \sigma_1^{\mathcal{C}}$ if ν is odd, and $\sigma_2^{\mathcal{C}} = -\sigma_1^{\mathcal{C}}$ if ν is even. ■

By similar reasoning, we can also show that for M_3 defined as $\begin{bmatrix} A & 0 \\ -I + F & B^T \end{bmatrix}$,

$$|M_1| = (-1)^{\nu(r+1)} |M_3|$$

Bibliography

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

[2] M. Barezzani, E. Pedrinelli, and M. Gerla. Protection planning in transmission networks. In Proceedings of the IEEE International Conference on Communications, volume 2, pages 316.4.1–316.4.5, 1992.

[3] Kenneth P. Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. ACM Transactions on Computer Systems, 17(2):41–88, 1999.

[4] N. Cai and R. W. Yeung. Network coding and error correction. Proceedings of IEEE Information Theory Workshop, pages 119–122, October 2002.

[5] N. Cai and R. W. Yeung. Network coding and error correction. Proceedings of IEEE Information Theory Workshop, pages 119–122, October 2002.

[6] N. Cai and R. W. Yeung. Secure network coding. In Proceedings. 2002 IEEE International Symposium on Information Theory, page 323, 2002.

[7] M. Castro and B. Liskov. Practical byzantine fault tolerance. In OSDI: Symposium on Operating Systems Design and Implementation. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, 1999.

[8] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing, October 2003.

[9] I. Csiszar. Linear codes for sources and source networks: Error exponents, universal coding. IEEE Transactions on Information Theory, 28, No.4:585–592, 1982.

[10] S. Deb and M. Médard. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. IEEE Transactions on Information Theory, April 2004, submitted.

[11] Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In Theory and Application of Cryptographic Techniques, pages 502–517, 2002.

[12] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. submitted to the IEEE Transactions on Information Theory, 2003.

[13] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. submitted to the IEEE Transactions on Information Theory, 2004.

[14] G. Ellinas and T. E. Stern. Automatic protection switching for link failures in optical networks with bi-directional links. In Proceedings IEEE GLOBECOM, 1996.

[15] G. Ellinas, T. E. Stern, and A. Hailemarian. Link failure restoration in optical networks with arbitrary mesh topologies and bi-directional links. Personal Communication, 1997.

[16] M. Feder, D. Ron, and A. Tavory. Bounds on linear codes for network multicast. Electronic Colloquium on Computational Complexity, 10(033), 2003.

[17] C. Fragouli, E. Soljanin, and A. Shokrollahi. Network coding as a coloring problem. 2004.

[18] T. Frisanco. Optimal spare capacity design for various protection switching methods in ATM networks. In Proceedings of the IEEE International Conference on Communications, volume 1, pages 293–298, 1997.

[19] J. Garay and Y. Moses. Fully polynomial byzantine agreement for n 3t processors in t+1 rounds. SIAM Journal of Computing, 27(1):247–290, February 1998.

[20] A. Gersht, S. Kheradpir, and A. Shulman. Dynamic bandwidth-allocation and path-restoration in SONET self-healing networks. IEEE/ACM Transactions on Networking, 45(2):321–331, June 1996.

[21] W.D. Grover. Case studies of survivable ring, mesh and mesh-arc hybrid networks. In Proceedings IEEE GLOBECOM, pages 633–638, 1992.

[22] W.D. Grover, T.D. Bilodeau, and B.D. Venables. Near optimal spare capacity placement in a mesh restorable network. In Proceedings IEEE GLOBECOM, pages 57.1.1–57.1.6, 1991.

[23] M. Herzberg and S.J. Bye. An optimal spare-capacity assignment model for survivable networks with hop limits. In Proceedings IEEE GLOBECOM, volume 3, pages 1601–1606, 1994.

[24] M. Herzberg, S.J. Bye, and A. Utano. The hop-limit approach for spare-capacity assignment in survivable networks. IEEE/ACM Transactions on Networking, pages 775–784, December, 1995.

[25] T. Ho, D. R. Karger, M. Médard, and R. Koetter. Network coding from a network flow perspective. In Proceedings of the IEEE International Symposium on Information Theory, 2003.

[26] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In Proceedings of 2003 IEEE International Symposium on Information Theory, June 2003.

[27] T. Ho, M. Médard, M. Effros, and R. Koetter. Network coding for correlated sources. In Proceedings of Conference on Information Sciences and Systems, 2004. To appear.

[28] T. Ho, M. Médard, and R. Koetter. A coding view of network capacity, recovery and management. In International Symposium on Information Theory, 2002.

[29] T. Ho, M. Médard, and R. Koetter. A coding view of network recovery and management for single-receiver communications. In Proceedings of the 2002 Conference on Information Sciences and Systems, 2002.

[30] T. Ho, M. Médard, and R. Koetter. An information theoretic view of network management. In Proceedings IEEE INFOCOM, 2003.

[31] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger. On randomized network coding. In Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing, October 2003.

[32] Tracey Ho, Ben Leong, Ralf Koetter, Muriel Médard, Michelle Effros, and David R. Karger. Byzantine modification detection in multicast networks using randomized network coding. In International Symposium on Information Theory, 2004.

[33] Tracey Ho, Ben Leong, Muriel Médard, Ralf Koetter, Yu-Han Chang, and Michelle Effros. On the utility of network coding in dynamic environments. In Proceedings of International Workshop on Wireless Ad Hoc Networks, 2004.

[34] S. Jaggi, P.A. Chou, and K. Jain. Low complexity algebraic network codes. In Proceedings of the IEEE International Symposium on Information Theory, 2003.

[35] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random nc. Combinatorica, 6 (1):35–48, 1986.

[36] K. P. Kihlstrom, L. E. Moser, and P. M. Melliar-Smith. The SecureRing protocols for securing group communication. In Proceedings of the 31st Annual Hawaii International Conference on System Sciences (HICSS), volume 3, pages 317–326. IEEE Computer Society Press, 1998.

[37] Murali S. Kodialam, T. V. Lakshman, and Sudipta Sengupta. Online multicast routing with bandwidth guarantees: a new approach using multicast network flow. In Measurement and Modeling of Computer Systems, pages 296–306, 2000.

[38] R. Koetter and M. Médard. An algebraic approach to network coding. In International Symposium on Information Theory, page 104, 2001.

[39] R. Koetter and M. Médard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, 2003.

[40] R. Koetter and M. Médard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, 11(5), October 2003.

[41] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4 (3):382–401, July 1982.

[42] April Rasala Lehman and Eric Lehman. Complexity classification of network information flow problems. In Symposium on Discrete Algorithms, 2004.

[43] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. IEEE Transactions on Information Theory, 49:371–381, 2003.

[44] Z. Li and B. Li. Network coding in undirected networks. 2004.

[45] D. Lun, M. Médard, and T. Ho. On network coding with a cost criterion. In Invited Paper, IEEE Communications Theory Workshop, year =.

[46] D. Malkhi and M. Reiter. A high-throughput secure reliable multicast protocol. Journal of Computer Security, 5:113–127, 1997.

[47] J. L. Massey. Contemporary cryptography: An introduction. Contemporary Cryptology: The Science of Information Integrity, pages 1–39, 1991.

[48] M. Médard, R.A. Barry, S.G. Finn, W. He, and S.S. Lumetta. Generalized loop-back recovery in optical mesh networks. IEEE/ACM Transactions on Networking, 10(1):153–164, February 2002.

[49] M. Médard, M. Effros, T. Ho, and D. R. Karger. On coding for non-multicast networks. In Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing, October 2003.

[50] R. Motwani and P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.

[51] N.H.M. Nizam, G.K. Hunter, and D.G. Smith. A dynamic reconfiguring tool for improving multiwavelength transport network robustness. In Proceedings of the IEEE International Conference on Communications, volume 1, pages 246–250, 1997.

[52] Taku Noguchi, Takahiro Matsuda, and Miki Yamamoto. Performance evaluation of new multicast architecture with network coding. IEICE Transactions on Communication, E86-B, No.6, June 2003.

[53] P. Papadimitratos and Z.J. Haas. Secure routing for mobile ad hoc networks, January 2002.

[54] R. Perlman. Network Layer Protocols with Byzantine Robustness. PhD thesis, Massachusetts Institute of Technology, October 1988.

[55] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part I - protection. In Proceedings IEEE INFOCOM, pages 744–751, 1999.

[56] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part II - restoration. In Proceedings of the IEEE International Conference on Communications, pages 2023–2030, 1999.

[57] S. Riis. Linear versus non-linear boolean functions in network flow, preprint, November 2003.

[58] H. Sakauchi, Y. Nishimura, and S. Hasegawa. A self-healing network with an economical spare-channel assignment. In Proceedings IEEE GLOBECOM, volume 1, pages 403.1.1–403.1.6, 1990.

[59] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In 15th ACM Symposium on Parallel Algorithms and Architectures, pages 286–294, 2003.

[60] Sergio D. Servetto. Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[61] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. IEEE Transactions on Information Theory, 25:471 – 480, 1973.

[62] B.D. Venables, W.D. Grover, and M.H. MacGregor. Two strategies for spare capacity placement in mesh restorable networks. In Proceedings of the IEEE International Conference on Communications, volume 1, pages 267–271, 1993.

[63] O.J. Wasem. An algorithm for designing rings for survivable fiber networks. IEEE Transactions on Reliability, 40, 1991.

[64] C.S. Wu, S.W. Lee, and Y.T. Hou. Backup vp preplanning strategies for survivable multicast ATM. In Proceedings IEEE GLOBECOM, pages 267–271, 1997.

[65] Y. Wu, P. A. Chou, and S.-Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. 2004, submitted.

[66] Y. Zhu, B. Li, and J. Guo. Multicast with network coding in application-layer overlay networks. IEEE Journal on Selected Areas in Communications, 22(1), 2004.