An information theoretic view of network management

Tracey Ho, Muriel Médard and Ralf Koetter

***Abstract*— We present an information theoretic framework for network management for non-ergodic link failures. Building on recent work in the field of network coding, we describe the input-output relations of network nodes as codes and quantify network management by the logarithm of the number of different codes needed for different failure scenarios. We give bounds on network 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. This is the first paper to our knowledge that looks at network management for general connections.**

I. Introduction

The problem of failing links in a network is well known. Various schemes to recover from link failures have been devised, among them live end-to-end path protection, loopback, and generalized loopback, which are used in different situations and have different advantages. What they 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. Reducing management overhead by minimizing the number of network states and affected nodes is desirable, other things being equal.

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. The framework which we consider is also independent of the specifics of circuit switched or packet switched networks.

Our approach has its roots in recent work on network coding [1], [2], [3], [4]. Ahlswede et al [2] showed that the traditional approach of transmitting information by routing or replication is not always sufficient to achieve maximum capacity for multicast, and that it may be necessary for to code together signals from different incoming links. Médard [3], [4] introduced a powerful algebraic framework for analyzing network coding. It is not yet clear how widely coding is needed to achieve capacity, but it is useful for robust recovery. In particular, [4] showed that with coding, a multicast network has a linear *receiver-based* (a term defined shortly) solution for all recoverable failures, i.e, a solution in which only the receiver nodes need to be informed of the failure pattern, while the other interior nodes need not change their behavior.

This leads to a very general concept of network behavior as a code, and provides a fundamental way to quantify essential management information as that needed to switch among different codes (behaviors) for different failure scenarios.

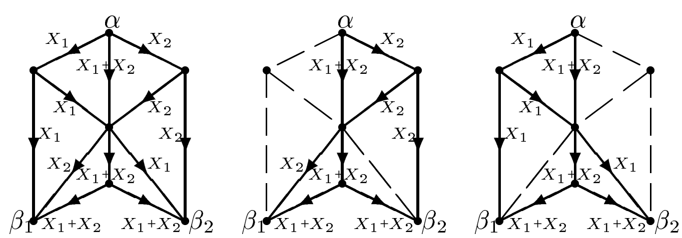

Fig. 1. An example of a receiver-based recovery scheme. Each diagram gives a code that is valid for failure of any of the links represented by dashed lines.

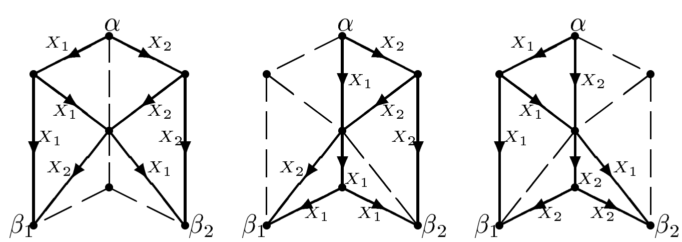

Fig. 2. An example of a network-wide code. Each diagram gives a code which is valid for failure of any of the links represented by dashed lines.

In this paper we analyze a *centralized* formulation for quantifying network management, in which the manage-

Tracey Ho and Muriel Médard are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, e-mail: {trace, medard}@mit.edu

Ralf Koetter is with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, e-mail: koetter@csl.uiuc.edu

ment requirement is taken as the logarithm of the number of codes that the network switches among.

As an illustration of some key concepts, consider the simple example network in Figure 1, in which the source node α simultaneously sends processes X_1 and X_2 to two receiver nodes β_1 and β_2. These connections can be recovered under failure of any one link in the network.

A receiver-based scheme is shown in Figure 1. Recovery schemes which involve any combination of receiver or interior nodes are called *network-wide* schemes. One possible set of codes for a network-wide recovery scheme is given in Figure 2. Note that 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.

It so happens that here the minimum centralized management requirement is log(3) in both cases, but we shall see that the number of codes for receiver-based and network wide recovery can differ in some cases.

This is the first work to our knowledge to consider general connections. This paper builds on work first begun in [5] and [6]. Reference [5] considered the multi-transmitter single-receiver case, and [6] considered the multi-transmitter multicast case, and presented results for failures of links adjacent to the receiver nodes.

Our main results provide, for network management information bits necessary to achieve link failure recovery over general networks, a lower bound for arbitrary connections and an upper bound for multi-transmitter multicast connections.

We present our model in Section II, state our main results in Section III and give the mathematical development, proofs and ancillary results in the remaining sections.

II. Model

As in [3], we represent a network by a directed graph $\mathcal{G}$ with vertices representing nodes and directed edges representing links. In this paper we consider delay-free acyclic networks. Discrete independent random processes $X(\alpha, 1), \ldots, X(\alpha, r_\alpha)$ are observable at one or more source nodes α, and processes originating at different source nodes are independent. There are one or more receiver nodes, comprising a set $\mathcal{D}$. The general network connection problem is to transmit a given subset $\mathcal{X}_\beta$ of the source processes to each receiver node $\beta \in \mathcal{D}$. The *multicast* connection problem is to transmit all the source processes to each of the receiver nodes.

Edge l is an *incident outgoing link* of node v if $v =$ tail(l), and an *incident incoming link* of v if $v =$ head(l). We call an incident incoming link of a receiver node a *terminal link*, and other links *interior links*. Edges l_1 and l_2 are *incident* if head(l_1) = tail(l_2) or head(l_2) = tail(l_1). Edge l carries the random process $Y(l)$. Output processes at a receiver node β are denoted $Z(\beta, i)$.

We choose the time unit such that the capacity of each link is one bit per unit time, and the random processes $X(\alpha, i)$ have a constant entropy rate of one bit per unit time. Edges with larger capacities are modelled as parallel edges, and sources of larger entropy rate are modelled as multiple sources at the same node.

The processes $X(\alpha, i)$, $Y(l)$, $Z(\beta, i)$ generate binary sequences. We assume that information is transmitted as vectors of bits which are of equal length u, represented as elements in the finite field $\mathbb{F}_{2^u}$. The length of the vectors is equal in all transmissions and all links are assumed to be synchronized with respect to the symbol timing.

We first consider linear coding, which has been shown by Li and Yeung [1] to be sufficient for multicast. In a linear code, the signal $Y(j)$ on an outgoing link j of a node $v =$ tail(j) is a linear combination of processes $X(v, n)$ generated at node v, and signals $Y(i)$ on incident incoming links i (ref Figure 3):

$$Y(j) = \sum_{i=1}^{r_v} A(i,j) X(v,i) + \sum_{i \,:\, \text{head}(i) = v} F(i,j) Y(i)$$

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

$$Z(\beta, i) = \sum_{j \,:\, \text{head}(j) = \beta} B_\beta(i,j) Y(j)$$

Expressing these equations in matrix form, we have:

$$\begin{bmatrix} Y(1) \\ \vdots \\ \vdots \\ Y(\mu) \end{bmatrix} = \begin{bmatrix} X(\alpha_1, 1) \cdots X(\alpha_n, r_n) \end{bmatrix} \begin{bmatrix} A(1,1) & \ldots & A(1,\mu) \\ \vdots & & \vdots \\ A(r,1) & \ldots & A(r,\mu) \end{bmatrix}$$

$$+ \begin{bmatrix} F(1,1) & \ldots & F(1,\mu) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ F(\mu,1) & \ldots & F(\mu,\mu) \end{bmatrix} \begin{bmatrix} Y(1) \\ \vdots \\ \vdots \\ Y(\mu) \end{bmatrix}$$

$$\begin{bmatrix} Z(\beta_1, 1) \cdots Z(\beta_{n'}, r_{n'}) \end{bmatrix} = \begin{bmatrix} Y(1) \\ \vdots \\ \vdots \\ Y(\mu) \end{bmatrix}^T \begin{bmatrix} B(1,1) & \cdots & B(r',1) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ B(\mu,1) & \cdots & B(r',\mu) \end{bmatrix}$$

Matrices A, F and B have coefficients in $\mathbb{F}_{2^u}$, and their structure is constrained by the network. A triple (A, F, B) specifies the behavior of the network, and represents a *linear network code*.

We assume a constant zero signal is observed on failed links, but discuss in Section IV-A the effect on network

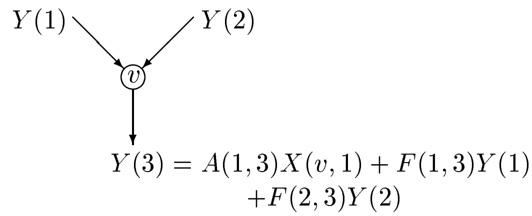

Fig. 3. Illustration of linear coding at a node.

management of not being able to make this assumption. 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.

III. Main results

We call a link h *integral* if it satisfies the property that there exists some subgraph of the network containing h, on which the set of source-receiver connections is feasible if and only if h has not failed.

Theorem 1—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.[1] □

This result shows that recovery using linear codes from all recoverable failures always requires network management.

The following theorems give bounds on the number of codes needed in different situations, 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;
- $t_{\min} = \min_\beta t_\beta$, the minimum number of terminal links among all receivers.

These bounds translate directly into bounds on the centralized network management requirement, by taking the logarithm of the number of codes. By tight bounds we mean that for any values of the parameters in terms of which the bounds are given, there are examples in which these bounds are met with equality.

[1]A solution with static A and F matrices always exists for any recoverable set of failures in a multicast scenario [3], but the receiver code B must change.

Theorem 2—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	$\begin{cases} \left\lceil \frac{m+1}{m-r+1} \right\rceil + 1 & \text{if } \left\lceil \frac{m+1}{m-r+1} \right\rceil \text{ integer and } d \geq 2 \\ \left\lceil \frac{m+1}{m-r+1} \right\rceil & \text{otherwise} \end{cases}$

□

Theorem 3—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	$\begin{cases} r+1 & \text{for } r = 1 \text{ or } m-1 \\ r & \text{for } 2 \leq r \leq m-2 \end{cases}$
network-wide	$\begin{cases} 2 & \text{for } r = 1 \\ r & \text{for } r = 2, 3, m-1 \\ r-1 & \text{for } 4 \leq r \leq m-2 \end{cases}$

b) For the **multicast case with two receivers**, an upper bound on the number of linear codes for the no-failure scenario and **all single link failures** is r^2+1.

c) For the **multicast case with** $d \geq 3$ **receivers**, an upper bound on the number of linear codes for the no-failure scenario and **all single link failures** is $(r+1)^d$.

d) For the **general case**, an upper bound on the number of linear codes for the no-failure scenario and all **single terminal link failures** is given by

$$\sum_{\beta : t_\beta \leq r} t_\beta + \sum_{\beta : t_\beta \geq r+1} r - 1$$

where the sum is taken over receiver nodes β.

□

Theorem 4—Nonlinear receiver-based recovery: For the multicast case, tight bounds on the number of nonlinear receiver-based codes for the no-failure scenario and terminal link failures are:

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

□

This result shows that if nonlinear processing is allowed, in some cases a single code suffices, i.e. network management is not required for recovery.

IV. Mathematical model

A linear network code is specified by a triple of matrices A, F and B, defined in Section II. The product $A(I-F)^{-1}B^T = AGB^T$ defines a transfer matrix from the source processes $\underline{X}$ to the output processes $\underline{Z}$ [3]. 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, and $G = I + F + F^2 + \dots$ sums the gains along all paths between each pair of links, and equals $(I - F)^{-1}$, since matrix F is nilpotent.

We use the following notation in this paper:

- $\underline{c}_j$ is column j of AG.
- $\underline{b}_j$ is column j of B.
- $\mathcal{T}_\beta$ is the set of terminal links of a receiver β.
- $G_\mathcal{K}$ is the submatrix of G consisting of columns that correspond to links in a set $\mathcal{K}$.
- $B_\mathcal{K}$ is the submatrix of B consisting of columns that correspond to links in a set $\mathcal{K}$.
- G_β is the submatrix consisting of columns $\underline{g}_j$ of G such that j is a terminal link of β.
- B_β is the submatrix of B consisting of columns $\underline{b}_j$ such that j is a terminal link of β.
- 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 $\mathcal{H}$.

An example illustrating the structure of the transfer matrices is given in Figure 4.

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 $AGB^T_\beta = \left[\underline{e}_{i_1} | \dots | \underline{e}_{i_R}\right]$, where $i_1, \dots, i_R$ are the elements of $\mathcal{X}_\beta$ in some specified order[2], and $\underline{e}_i$ is the unit column vector whose only nonzero entry is in the i^{th} position. In the single-receiver and multicast cases, we choose the same ordering for input and output processes, so this condition becomes $AGB^T_\beta = I \ \forall \ \beta$. 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 $AI^hG^h(BI^h)^T$, 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 $G^h = I^h + F^h + (F^h)^2 + \dots = I^h\left(I - FI^h\right)^{-1} = \left(I - I^hF\right)^{-1} I^h$. Note that $AI^hG^h(BI^h)^T = AG^hB^T$ since $I^hI^h = I^h$. If failure of link h is recoverable, there exists some (A', G', B') such that for all $\beta \in \mathcal{D}$, $A'G'^hB'^T_\beta = \left[\underline{e}_{i_1} | \dots | \underline{e}_{i_R}\right]$ where $\mathcal{X}_\beta = \{i_1, \dots, i_R\}$.

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

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

A. Codes for multiple failure scenarios

We first characterize codes which can be used for multiple single link failures, developing concepts and tools used in later sections.

A path is a sequence of distinct nodes that are connected by links. 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.

Lemma 1: Let $\mathcal{T}^h_\beta \subseteq \mathcal{T}$ be the set of terminal links of each receiver β that are downstream of link h.
1. If code (A, G, B) covers the no-failure scenario and failure of link h, then $\underline{c}_h \ \sum_{j\in\mathcal{T}^h_\beta} G(h,j)\underline{b}^T_j = \mathbf{0} \ \ \forall \ \beta \in \mathcal{D}$.
2. If code (A, G, B) covers failures of links h and k, then $\forall \ \beta \in \mathcal{D}$, either

(a) $\quad \underline{c}_h \sum_{j\in\mathcal{T}^h_\beta} G(h,j)\underline{b}^T_j = \mathbf{0}$
and $\quad \underline{c}_k \sum_{j\in\mathcal{T}^k_\beta} G(k,j)\underline{b}^T_j = \mathbf{0}$
or
(b) $\quad \gamma_{h,k} \sum_{j\in\mathcal{T}^h_\beta} G(h,j)\underline{b}^T_j = \sum_{j\in\mathcal{T}^k_\beta} G(k,j)\underline{b}^T_j \neq \mathbf{0}$
and $\quad \underline{c}_h = \gamma_{h,k}\underline{c}_k \neq \mathbf{0}$
where $\gamma_{h,k} \in \mathbb{F}_{2^u}$ is a constant for given h, k

Proof outline: The results follow from writing $AG^h_\mathcal{T}B^T_\mathcal{T}$ in the form $\sum_{j\in\mathcal{T}} \underline{c}^h_j \underline{b}^T_j$ and noting that $\Delta\underline{c}^h_j = \underline{c}_j - \underline{c}^h_j = G(h,j)\underline{c}_h$. ∎

These expressions simplify considerably for terminal links as follows:

Corollary 1: 1. If code (A, G, B) covers the no-failure scenario and failure of terminal link h, then $\underline{c}_h\underline{b}^T_h = \mathbf{0}$.
2. If (A, G, B) covers failures of two terminal links h and k, then either

(a) $\quad \underline{c}_h\underline{b}^T_h = \mathbf{0}$ and $\underline{c}_k\underline{b}^T_k = \mathbf{0}$
or
(b) $\quad h$ and k are terminal links of the same receiver β,
$\gamma_{h,k}\underline{b}^T_h = \underline{b}^T_k \neq \mathbf{0}$ and $\underline{c}_h = \gamma_{h,k}\underline{c}_k \neq \mathbf{0}$
where $\gamma_{h,k} \in \mathbb{F}_{2^u}$ is a constant for given h, k

These results lead to the notion of active and non-active recovery codes. A recovery code which is *active* in a failed link h is one in which AG^hB^T is affected by the value on link h, i.e. $\underline{c}_h \sum_{j\in\mathcal{T}^h_\beta} G(h,j)\underline{b}^T_j \neq \mathbf{0}$ for some receiver β. Otherwise, the code is *non-active* in h. Active codes cannot be used if signals on failed links are undetermined.

In a code which is non-active in a failed link, the value on that link is set to zero (by upstream links ceasing to

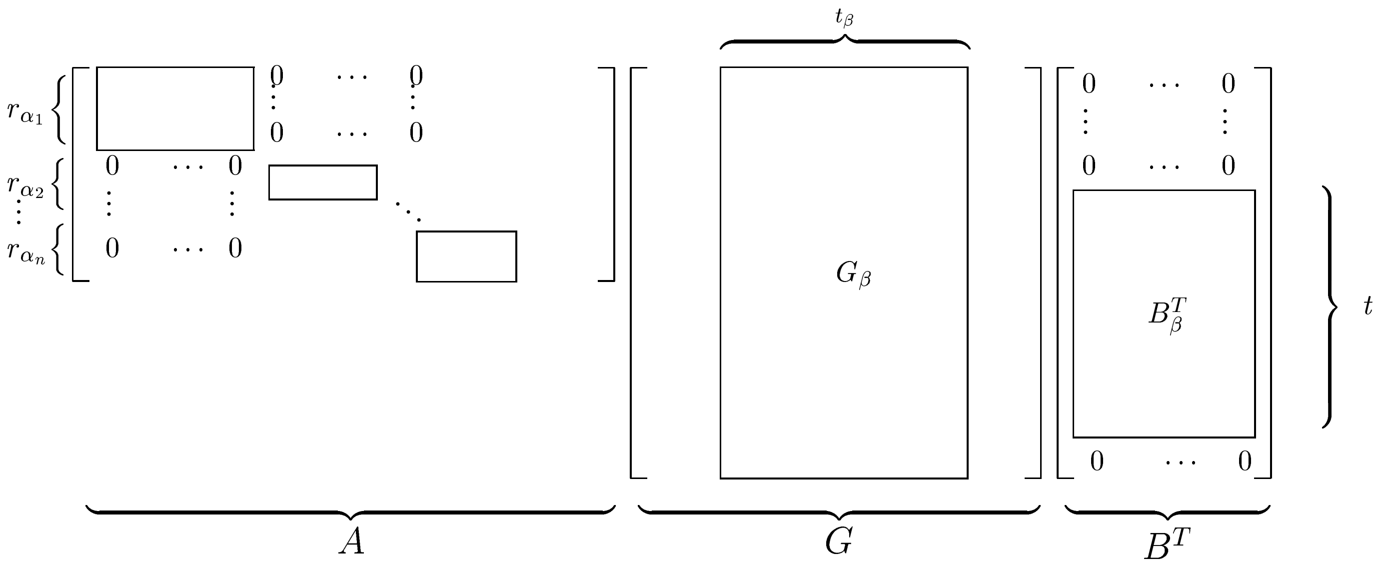

Fig. 4. An example illustrating the structure of transfer matrices.

transmit on the link), cancelled out, or disregarded by the receivers. Part 1 of Lemma 1 states that a code which covers the no-failure scenario as well as one or more single link failures must be non-active in those links. Part 2 of Lemma 1 states that a code which covers failures of two or more single links is 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.

V. Need for network management

Proof of Theorem 1: 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 1, 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. ■

VI. Bounds on linear network management requirement

A. Single receiver analysis

Let $\mathcal{M}$ be a set of links on a minimum capacity cut between the sources and the receiver[3], where $|\mathcal{M}| = m$.

[3]a partition of the network nodes into a set containing the sources, and another set containing the receiver, such that the minimum number of links cross from one set to the other

We define transfer matrices for the set $\mathcal{J}$ of links upstream of and including links in $\mathcal{M}$. Let (Q, J) be a partial interior code defining the behavior of links in $\mathcal{J}$, where

1) $r \times |\mathcal{J}|$ matrix Q specifies how the source processes X_i, $i = 1, \dots, r$ are represented on the source nodes' incident outgoing links[4]. The signal on source link j is $Y(j) = \sum_{i=1}^{\rho} A(i,j)X_i$, where X_i are processes generated at tail(j) and transmitted on j.
2) $|\mathcal{J}| \times |\mathcal{J}|$ matrix D specifies how signals are transmitted between incident links in $\mathcal{J}$. $D(i,j)$ is nonzero only if head(i) = tail(j). If head(i) = tail(j) for links $i = 1, \dots, n$ and link j, the signal on link j is $Y(j) = \sum_{i=1}^{n} D(i,j)Y(i)$.
3) $|\mathcal{J}| \times |\mathcal{J}|$ matrix $J = I + H + H^2 + \dots$ sums the gains along all paths between each pair of links, and equals $(I - H)^{-1}$ since matrix H is nilpotent.

We define the following:

- $J_{\mathcal{M}}$ is the submatrix of J consisting of columns that correspond to links in $\mathcal{M}$.
- $A_{\mathcal{J}}$ is the submatrix of A consisting of columns that correspond to links in $\mathcal{J}$.
- $G_{\mathcal{J}\times\mathcal{J}}$ is the $|\mathcal{J}| \times |\mathcal{J}|$ submatrix of G consisting of rows and columns corresponding to links in set $\mathcal{J}$.
- Π' is a related network connection problem in which all nodes upstream of $\mathcal{M}$, and the links between them, are the same as the original problem, but each link h in $\mathcal{M}$ is replaced by a link h' such that tail(h') = tail(h), and head(h') = β', a new receiver node that

[4]all of which are in $\mathcal{J}$, since $\mathcal{M}$ is a cut between the source nodes and the receiver

is the head of all links h'.

Note that Q, D and J are defined analogously to A, F and G respectively, except that they are limited to specifying the behavior of links in $\mathcal{J}$. $A_{\mathcal{J}}$ is a value for Q that corresponds to A, and $G_{\mathcal{J}\times\mathcal{J}}$ is a value for Q that corresponds to G.

The following two lemmas allow us to relate codes for terminal link failures in problem Π' to codes for failures of links in $\mathcal{M}$.

Lemma 2: 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 $QJ^h_{\mathcal{M}}L^T = I$ for $h \in \mathcal{M}_1 \subseteq \mathcal{M}$, 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 $QJ^h_{\mathcal{M}}L^T = I$ for $h \in \mathcal{M}_1$.

Proof outline: There exists a set of link-disjoint paths $\{P_k \mid k \in \mathcal{M}\}$ where P_k connects link k to the receiver. (Q, J) can be *extended* 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 without coding along the path P_k to the receiver. For the converse, we can construct a matrix L which satisfies the required property as follows:

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

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

Lemma 3: 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 outline: 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. By Lemma 2, a valid partial interior code (Q, J) in which no link in $\mathcal{M}$ feeds into another can be extended to a valid code (A, G). ■

Lemma 4: 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 terminal link failures is

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

Proof outline: 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 $t = r + 1$, $\left\lceil \frac{t}{t-r} \right\rceil = t$. We can cover each of the $r + 1$ terminal links by a separate code, so $r + 1$ codes suffice. For $2 \le r \le t - 2$, 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. We can find two pairs $(\underline{v}_i, \underline{w}_{i'})$ and $(\underline{v}_j, \underline{w}_{j'})$ such that $\underline{w}_{i'}$ can replace $\underline{v}_i$ in the basis, and $\underline{w}_{j'}$ can replace $\underline{v}_j$ in the basis. 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 $\{\underline{w}_k \mid k = 1, \ldots, t-r,\ k \ne i', j'\}$ by another code, and the links corresponding to $\{\underline{v}_k | k = 1, \ldots, r, k \ne i, j\}$ by a separate code each. ■

Lemma 5: For any set of $n \ge 2$ 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 maps which are multiples of each other. One of the links can be arbitrarily designated as the primary for the set. If all n codes are active codes which cover two or more terminal link failures, then only $2 \le n$ 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 primary link of each active code can be covered together with some non-active code, and its secondary links can be covered by a new non-active code. This forms a set of n non-active codes covering the same terminal link failures as the original set. ■

Corollary 2: For receiver-based recovery, the minimum number of codes for terminal link failures can be achieved with non-active codes.

Lemma 6: 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
$\left\lceil \frac{m}{m-r} \right\rceil$	$\max\left(\left\lceil \frac{m}{m-r} \right\rceil, r\right) = \begin{cases} r+1 & \text{for } r = 1 \text{ or } m-1 \\ r & \text{for } 2 \le r \le m-2 \end{cases}$

Proof outline: It follows from Lemma 3 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 2). The upper bound from Lemma 4 thus applies, with m in place of t.

For the lower bound, from Lemma 1, a single code in a valid receiver-based scheme can cover at most $m - r$

of the links in $\mathcal{M}$. By Corollary 2, restricting consideration to non-active codes does not increase the receiver-based lower bound for the related terminal link problem Π', which is also $\left\lceil \frac{m}{m-r} \right\rceil$, and so does not increase the receiver-based lower bound here. ■

Lemma 7: 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 $\left\lceil \frac{m+1}{m-r+1} \right\rceil$.

Proof outline: It follows from Lemma 1 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 8: 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 outline: From Lemma 3, there exist valid codes for failures of links in $\mathcal{J}$ in problem Π'. Thus, a static interior code (Q', J') covering these failures exists for Π' [3]. This can be extended (Lemma 2) to a static interior code (A, G) in which no link in $\mathcal{M}$ feeds into another. For any such code (A, G), consider any link h which feeds into more than one link in $\mathcal{M}$. Let the set of these links be $\mathcal{M}^h = \{h_1, \ldots, h_x\}$, and let the set of remaining links in $\mathcal{M}$ be $\overline{\mathcal{M}^h}$.

Case 1: h feeds into some link h_i in $\mathcal{M}$ via some path P without further coding with other signals. We can construct a partial code (Q, J) in which h feeds only into $h_i \in \mathcal{M}^h$, whose extension is a valid static code.

Case 2: Coding occurs between h and each $h_i \in \mathcal{M}^h$. We 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 $\mathcal{M}^h$. Let h_j be some link in $\mathcal{M}^h \cap \mathcal{M}/\mathcal{L}$.

Case 2a: There exists a set R of links forming a single path from h to h_j, excluding h and h_j, such that none of the links $h' \in R$ feeding into some other link $h_i, i = 1, \ldots, x$, $i \neq j$ has a signal map other than $G(h, h')\underline{c}_h$. We can then construct a partial code (Q', D') which is the same as $(A_{\mathcal{J}}, F_{\mathcal{J}\times\mathcal{J}})$ except that h feeds only into links in R, whose extension is a valid static code.

Case 2b: Every path from h to h_j contains some link that feeds into one or more links $h_i \in \mathcal{M}^h$ besides h_j, and has a signal map which is a linear combination of $\underline{c}_h$ and some other signal map. Consider any path R' from h to h_j and let $\tilde{h}$ be the furthest upstream of these links.

We apply the entire argument described from paragraph 1 onwards with (A, G) and $\tilde{h}$. If case 1 or case 2a applies, then we have a modified code (A', G') in which $\tilde{h}$ feeds into only one link in $\mathcal{M}$. We then apply the same argument once again, this time to (A', G') and h, with h feeding into strictly fewer links in $\mathcal{M}$ than before. If on the other hand case 2b applies, we proceed recursively, with $\tilde{h}$ replaced by one of its downstream links. If we come to a link that is incident to a link in $\mathcal{M}$, then case 1 or case 2a will apply, allowing us to eliminate a nonzero number of links in $\mathcal{M}$ from consideration. Thus, the procedure terminates with a valid static interior code in which h feeds into only one link in $\mathcal{M}$. ■

Proof of Theorem 3a:

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 (Lemmas 2 and 8). A non-active code (A, G, B) which covers failure of link k also covers failure of all links in S_k and P_k. Thus the bounds for receiver-based, or static, recovery here are the same as those in Lemma 6. 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 5.

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

For $r = m - 1$, by Lemma 8, there exists a valid static interior code (A, G) such that no link feeds into more than one link in $\mathcal{M}$. Choose any link $h \in \mathcal{M}$ and let the set of remaining links in $\mathcal{M}$ be $\overline{\mathcal{M}^h}$. Consider any i such that $AG(i, h)$ is nonzero, i.e. link h carries signal i. Let $\underline{e}_i \in \mathbb{F}^r_{2^u}$ be the unit vector which has 1 in the i^{th} position as its only nonzero entry. Since no link feeds into more than one link in $\mathcal{M}$, column AG_h can be set to $\underline{e}_i$ without affecting any of the other columns in $AG_{\mathcal{M}}$. Since (A, G) is a valid static interior code covering failure of h, $AG_{\overline{\mathcal{M}^h}}$ has full rank, so $\underline{e}_i$ is a linear combination of some subset $\mathcal{M}^h_i$ of columns in $\mathcal{M}^h$. There exists some $k \in \mathcal{M}^h_i$ for which $AG(i, k)$ is nonzero. Column AG_k can be set to $\underline{e}_i$ without reducing the rank of $AG_{\mathcal{M}}$, since AG_k is a linear combination of the other columns in $\mathcal{M}^h_i$, together with $\underline{e}_i$. Then h and k and their upstream links can be covered by a single active code. The remaining $r - 1$ links in $\mathcal{M}$, and their upstream links, can be covered by their corresponding receiver-based codes. An example in which $r = m - 1$, and r network-wide codes are needed is given in Figure 6.

For $4 \leq r \leq m-2$, we start with a static interior code in which no link feeds into more than one link in $\mathcal{M}$. We can

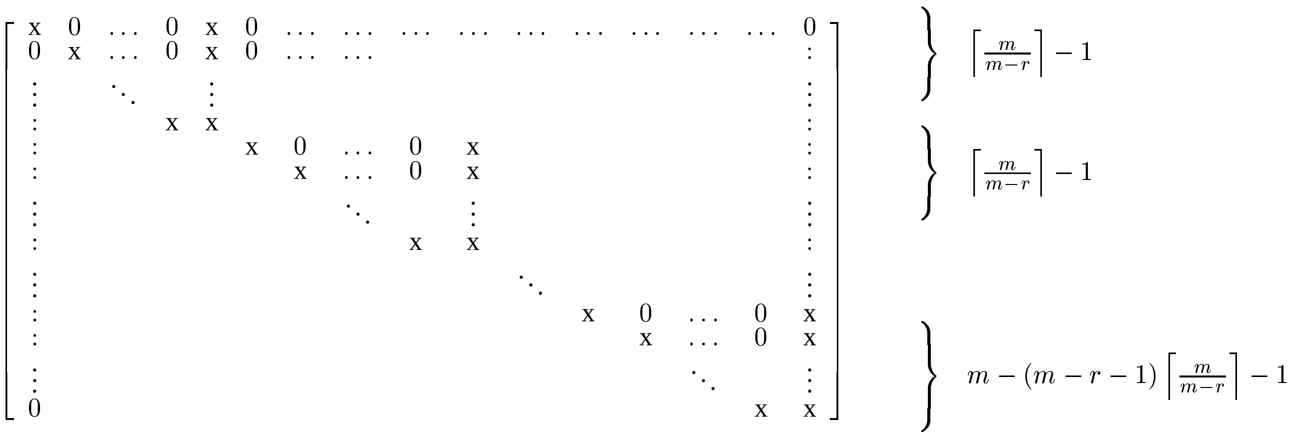

Fig. 5. An example of an $AG_{\mathcal{M}}$ matrix in a receiver-based code that achieves the lower bound of $\left\lceil \frac{m}{m-r} \right\rceil$ codes, with $m - (m - r - 1)\left\lceil \frac{m}{m-r} \right\rceil \geq 2$.

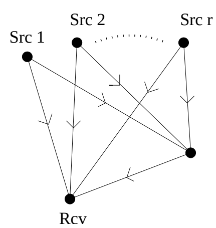

Fig. 6. An example network in which $r = m - 1$, which achieves the linear receiver-based upper bound of $r + 1$ codes and the linear network-wide and nonlinear receiver-based upper bounds of r codes.

modify the code to remove pairwise dependency among columns, except in cases where the dependent columns are from a set of 2^x links in $\mathcal{M}$ that are downstream only of the same x processes.

Case 1: There is a set of $r + 2$ columns in $AG_{\mathcal{M}}$ which contains a basis and such that no two columns of the set are multiples of each other. Then the set contains three pairs of columns such that each pair can be covered by a single code, and $r+2-3 = r-1$ non-active codes suffice.

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. A pair of dependent columns is from a set of 2^x links in $\mathcal{M}$ that are downstream only of a set of x processes. Links in such a set can be covered by two non-active codes. Let ρ be the total number of processes involved in such sets. Then links which are not part of such sets can be covered by $r - \rho$ codes including $r - \rho - 1$ non-active codes. If $r - \rho - 1 \geq 2$, then $r - \rho$ codes suffice. If $r - \rho - 1 = 1$, then $r - \rho + 1 = 3 \leq r - 1$ codes suffice.

For $1 \leq r \leq 3$, the receiver-based upper bound of $\max(2, r)$ is also a tight upper bound for network-wide recovery, which includes the former as a special case.

The example network of Figure 7 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 $4 \leq r \leq m - 2$. ■

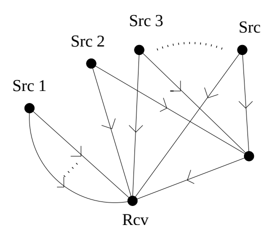

Fig. 7. An example network which achieves the receiver-based upper bound of r, the network-wide upper bounds of r codes for $r = 3$, and $r - 1$ codes for $4 \leq r \leq m - 2$.

B. General case lower bound

Proof of Theorem 2: 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 6 and 7). 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 8, where all terminal links can be covered by two non-active codes covering at the same time other interior links. This example also achieves the network-wide lower bound with equality when $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is not an integer. For two or more receivers, the network-wide lower bound of $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is not achievable when $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is an integer. This is because covering links on the minimum cut with exactly $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ codes would require $\left\lceil \frac{m+1}{m-r+1} \right\rceil - 1$ of them to be active codes, each covering $m-r+1$ links, and one of them to be non-active (Lemma 7). An acstive code cannot cover terminal link failures of two or more receivers (Corollary 1). Since at least two non-active codes are needed to cover terminal links of any receiver, $\left\lceil \frac{m+1}{m-r+1} \right\rceil - 1$ active codes and one non-active code are insufficient to cover terminal link failures for more than one receiver. If $\left\lceil \frac{m+1}{m-r+1} \right\rceil$ is not an integer, let $\left\lceil \frac{m+1}{m-r+1} \right\rceil (m-r+1) = m+1+y$. Then the number of non-active active codes $\min\left(\left\lceil \frac{m+1}{m-r+1} \right\rceil, y+1\right) \geq 2$, and the example network of Figure 8 achieves the network-wide lower bound of $\left\lceil \frac{m+1}{m-r+1} \right\rceil$.

■

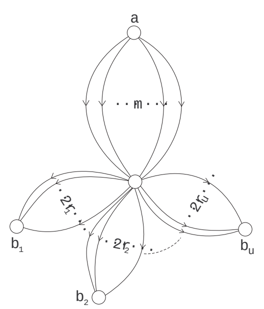

Fig. 8. An example network which achieves the general case lower bounds of Theorem 2 with equality, where r_i is the number of processes received by receiver β_i.

C. Upper bounds for all link failures, multicast case

Proof of Theorem 3c: From the proof of Theorem 2b, we know that for each receiver node β individually, there is a static solution for all single link failures in which m_β link-disjoint subgraphs feed into m_β different terminal links 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*. These trees can be grouped into $s_\beta \leq r+1$ link-disjoint forests 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. We will denote trees rooted at receiver β_x by $\mathcal{G}_x^i$, $i = 1, 2, \ldots$.

In the multicast case, if a network satisfies the max-flow min-cut condition for each receiver [2], then the connections to all receivers are simultaneously feasible. Thus a set of links intersecting 0 or 1 of these forests for each receiver can be covered together.

Let d be the number of receivers. Each of the $s_{\beta_1} \leq r+1$ forests for a receiver β_1 may contain links that are part of $\leq r+1$ such sets for receiver β_2, which have to be covered separately. Each of the resulting $\leq (r+1)^2$ subsets may in turn contain links that are part of $\leq r+1$ such sets for receiver β_3, and so on. Thus $\leq (r+1)^d$ codes are required. ■

Proof of Theorem 3b: Here we consider the two receiver case. The max flow min cut condition translates into the existence of a basis for all r processes among the signals on the trunks of each receiver's trees. If a receiver has more than $r+1$ trees, then these can be grouped into $\leq r$ forests which can each be covered together. If this is the case for both receivers, then at most r^2 codes are needed. If not, then at least one of the receivers, say β_1, has $r+1$ trees, any r of whose trunks carry signals forming a basis.

Let a link that lies on two trees $\mathcal{G}_1^i$ and $\mathcal{G}_2^j$ be called an *intersection*, denoted ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$). Intersections between the same two trees, $\mathcal{G}_1^i$ and $\mathcal{G}_2^j$, that form a contiguous path are considered part of the same intersection, and if they do not form a contiguous path but are not separated along both $\mathcal{G}_1^i$ and $\mathcal{G}_2^j$ by intersections involving other paths, then they are also considered part of the same intersection.

Suppose there exists an intersection of a tree $\mathcal{G}_1^i$ along one of its branches $\mathcal{B}_1^i$ with a tree $\mathcal{G}_2^j$. The trunks of the r trees other than $\mathcal{G}_1^i$ carry signals forming a basis, and the subtree of $\mathcal{G}_1^i$ excluding branch $\mathcal{B}_1^i$ can replace some tree $\mathcal{G}_1^{i'}$ in this basis. Then the intersection ($\mathcal{B}_1^i$, $\mathcal{G}_2^j$) can be covered together with intersections ($\mathcal{G}_1^{i'}$,$\mathcal{G}_2^j$), if any. A similar argument holds for an intersection of a tree $\mathcal{G}_2^{j'}$ along one of its branches with a tree $\mathcal{G}_1^{i''}$. We thus restrict our attention to the set of intersections ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) that involve only links on the trunks of the trees.

Suppose an intersection ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) is the furthest upstream intersection in $\mathcal{J}$ of some tree $\mathcal{G}_2^j$. Then there exists a set of r paths satisfying the max flow min cut condition between the sources and receiver β_1, that excludes the portion of the trunk of $\mathcal{G}_1^i$ upstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) and the trunk of one other tree of β_1. To see this, note that the trunks of the r trees other than $\mathcal{G}_1^i$ carry signals forming a basis. If $\mathcal{G}_2^j$ does not have any intersections upstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) with branches of other trees $\mathcal{G}_1^{i'}$, then joining the portion of $\mathcal{G}_1^i$ downstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) with the portion of $\mathcal{G}_2^j$ upstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) gives a tree which can replace one of the trees in the basis set. If $\mathcal{G}_2^j$ does have one or more intersections upstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) with branches of other trees $\mathcal{G}_1^{i'}$, let its furthest downstream of these intersections be with a branch $\mathcal{B}_1^{\tilde{i}}$ of tree $\mathcal{G}_1^{\tilde{i}}$. Consider the path formed by joining the portion of $\mathcal{B}_1^{\tilde{i}}$ upstream of this intersection with the portion of $\mathcal{G}_2^j$ between this intersection and ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$), and the portion of $\mathcal{G}_1^i$ downstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$). This path can replace one of the trees in the original basis set.

Let $\mathcal{G}_1^{\hat{i}}$ be the tree that is replaced in the basis set, i.e. its trunk is not part of the set of subgraphs satisfying the max flow min cut condition between the sources and receiver β_1. Then any intersection ($\mathcal{G}_1^i$, $\mathcal{G}_2^{j'}$) upstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) on $\mathcal{G}_1^i$ can be covered together with intersections ($\mathcal{G}_1^{\hat{i}}$, $\mathcal{G}_2^{j'}$) in $\mathcal{J}$, if any. Thus, we need only consider intersections along $\mathcal{G}_1^i$ that are downstream of ($\mathcal{G}_1^i$, $\mathcal{G}_2^j$) inclusive.

Let $\mathcal{I}$ be the set of all such intersections. Then the furthest upstream intersection in $\mathcal{I}$ of any tree $\mathcal{G}_1^i$ is with the furthest upstream intersection in $\mathcal{I}$ of some tree $\mathcal{G}_2^j$, and $\mathcal{I}$ contains intersections involving at most $r+1$ trees $\mathcal{G}_2^j$.

Suppose more than r^2+1 codes are needed.

Case 1: Each of the trees $\mathcal{G}_1^i$ has ≥ 2 intersections in $\mathcal{I}$. Then we can define an alternative set of disjoint trees $\mathcal{G}_2^{'j}$ such that one of the trees $\mathcal{G}_1^i$ has 0 or 1 intersection in $\mathcal{I}'$, where $\mathcal{I}'$ is defined similarly to $\mathcal{I}$, but with the alternative trees $\mathcal{G}_2^{'j}$ in place of the original trees $\mathcal{G}_2^j$. This puts us in case 2.

To show this, we consider the set K_1 of furthest upstream intersections of trees $\mathcal{G}_1^i$ in $\mathcal{I}$, and the set K_2 of second furthest upstream intersections of trees $\mathcal{G}_1^i$ in $\mathcal{I}$. Each intersection in K_1 is with a different tree $\mathcal{G}_2^j$, but there may be more than one intersection in K_2 with the same tree $\mathcal{G}_2^j$.

If there exists a subset of trees $\mathcal{G}_1^i$, $i \in S$ such that their intersections in K_2 are with the same set of trees $\mathcal{G}_2^j$ as their intersections in K_1, then we can define $\mathcal{G}_2^{'j}$, $j \in S$ to match the portion of the paths $\mathcal{G}_1^i$ between their first and second intersections in $\mathcal{I}$, as shown in Figure 9.

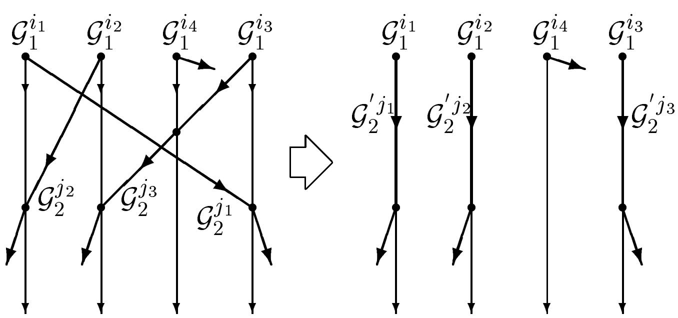

Fig. 9. Illustration of algorithm for defining trees $\mathcal{G}_2^{'j}$.

The new set of trees $\mathcal{G}_2^{'j}$ can be obtained with the following algorithm. Each $\mathcal{G}_2^{'j}$ is initialized to be the same as $\mathcal{G}_2^j$. We start with an intersection ($\mathcal{G}_1^{i_1}$, $\mathcal{G}_2^{'j_1}$) that is the furthest upstream in $\mathcal{I}$ for $\mathcal{G}_1^{i_1}$ and $\mathcal{G}_2^{'j_1}$. Let the adjacent downstream intersection for $\mathcal{G}_1^{i_1}$ be ($\mathcal{G}_1^{i_1}$, $\mathcal{G}_2^{'j_2}$), let the furthest upstream intersection in $\mathcal{I}$ for $\mathcal{G}_2^{'j_2}$ be ($\mathcal{G}_1^{i_2}$, $\mathcal{G}_2^{'j_2}$), and let the adjacent downstream intersection for $\mathcal{G}_1^{i_2}$ be ($\mathcal{G}_1^{i_2}$, $\mathcal{G}_2^{'j_3}$). If $\mathcal{G}_2^{'j_3} = \mathcal{G}_2^{'j_1}$, then the subset $\{\mathcal{G}_1^{i_1}, \mathcal{G}_1^{i_2}\}$ has intersections in K_1 and K_2 involving the same trees $\mathcal{G}_2^{'j_1}$ and $\mathcal{G}_2^{'j_2}$. We can redefine the portion of $\mathcal{G}_2^{'j_1}$ downstream of ($\mathcal{G}_1^{i_2}$, $\mathcal{G}_2^{'j_2}$) to match the portion of the paths $\mathcal{G}_1^{i_1}$ and $\mathcal{G}_1^{i_2}$ between their first and second intersections. This collapses the four intersections into two. If not, we continue in a similar fashion, letting the furthest upstream intersection in $\mathcal{I}$ for $\mathcal{G}_2^{'j_n}$ be ($\mathcal{G}_1^{i_n}$, $\mathcal{G}_2^{'j_n}$), and letting the adjacent downstream intersection for $\mathcal{G}_1^{i_n}$ be ($\mathcal{G}_1^{i_n}$, $\mathcal{G}_2^{'j_{n+1}}$), until $\mathcal{G}_2^{'j_{n+1}} = \mathcal{G}_2^{'j_p}$ for some $p < n+1$. Then the subset $\{\mathcal{G}_1^{i_p}, \dots \mathcal{G}_1^{i_n}\}$ has intersections in K_1 and K_2 involving the same trees $\{\mathcal{G}_2^{'j_p}, \dots \mathcal{G}_1^{i_n}\}$ and we can define $\mathcal{G}_2^{'j_p}, \dots, \mathcal{G}_2^{'j_n}$ to match the portion of the paths $\mathcal{G}_1^{i_p}, \dots, \mathcal{G}_1^{i_n}$ between their first and second intersections, collapsing $2(n-p+1)$ intersections to $n-p+1$ intersections. We repeat the process, redefining paths $\mathcal{G}_2^{'j}$ until no further redefinition is possible. As long as each path $\mathcal{G}_1^i$ has at least two intersections, carrying out this process always results in redefinition of paths $\mathcal{G}_2^{'j}$ to reduce the number of intersections. When no further redefinition is possible, there will be some path $\mathcal{G}_1^i$ that has 0 or 1 intersection in $\mathcal{I}$.

Case 2: Some tree $\mathcal{G}_1^i$ has 0 or 1 intersection in $\mathcal{I}$. Then $r+1$ trees $\mathcal{G}_2^j$ are involved in intersections in $\mathcal{I}$, and each of them has ≥ 2 intersections in $\mathcal{I}$. By similar reasoning as in Case 1, we can define an alternative set of disjoint trees $\mathcal{G}_1^{'i}$ such that one of the trees $\mathcal{G}_2^j$ has ≤ 1 intersection in $\mathcal{I}''$, where $\mathcal{I}''$ is defined similarly to $\mathcal{I}$, but with the alternative trees $\mathcal{G}_1^{'i}$ in place of the original trees $\mathcal{G}_1^i$.

■

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 10. 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 β_1 intersects one path to β_2, the second path to β_1 intersects two paths to β_2, the third intersects three and so on. Each of the $(r+1)(r+2)/2$ intersections requires a separate code.

The general 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 re-

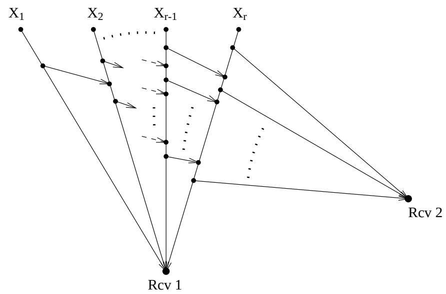

Fig. 10. An example multicast problem in which $(r+1)(r+2)/2$ codes are needed for all link failures.

quirement 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 general case, the number of codes for terminal link failures can grow linearly in the number of receivers.

Proof of Theorem 3d: 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.

Consider a receiver β that has $\geq r+1$ terminal links, and any interior code valid under failure of some other receivers' terminal links. 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. So at most $r-1$ of its terminal links terminal links cannot be covered together with any valid combination of terminal link failures of other receivers. ■

We have not yet determined whether this bound is tight. Figure 11 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.

VII. NONLINEAR RECEIVER-BASED RECOVERY

Proof of Theorem 2d: 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 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. However, this is not possible for $r > 1$ as there are only $q-1$ possible values for that coordinate. There will be at least r different codewords which give the same received codeword for different failures. For $r = 1, t = 2$, it is possible to satisfy this condition. 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 4 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 12. 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 failure scenario, 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 6.

Next we consider the multiple receiver case. We refer to the code generated by AG_β as a β code, and the code-

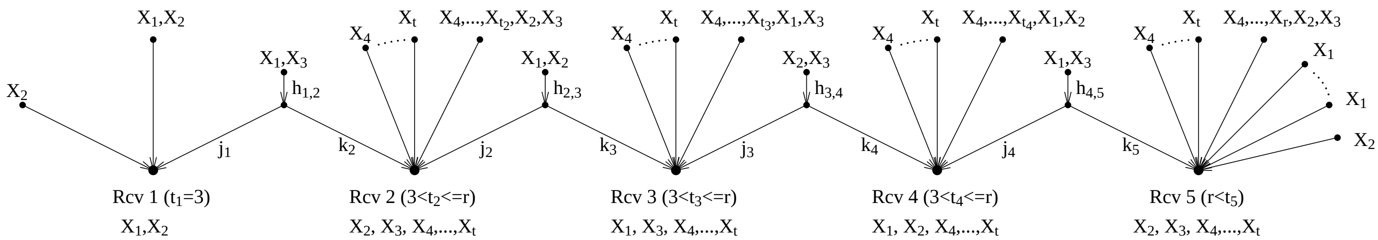

Fig. 11. An example network in which $\sum_{t_\beta \leq r} t_\beta - 2 + \sum_{t_\beta \geq r+1} r - 1$ codes are needed.

words 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. ■

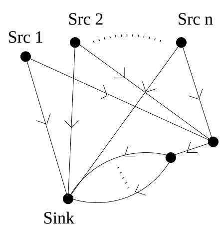

Fig. 12. An example network in which $2 \leq r \leq t - 2$, which achieves the nonlinear receiver-based upper bound of r codes.

VIII. Conclusions and further work

As the complexity of networks increases, so do the network management overhead and the catastrophic effects of imperfect network management. It is thus useful to understand network management in a fundamental way. We have proposed a framework for considering and quantifying network management, seeking through our abstraction not to replace implementation, but to guide it.

We have given a framework for quantifying network management in terms of the number of different network behaviors, or codes, required under different failure scenarios, and have provided bounds on network management requirements for various network connection problems in terms of basic parameters including the number of source processes, the number of links in a minimum source-receiver cut, and the number of terminal links.

There is much scope for future work in this area. One good area for further research 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. Other directions for further work include extending our results to networks with multiple receivers, non-multicast connections and cycles and delay, studying the capacity required for transmission of network management signals, and considering network management for wireless networks.

References

[1] S.-Y.R. Li and R.W. Yeung, “Linear Network Coding”, preprint, 1999.

[2] R. Ahlswede, N. Cai, S.-Y.R. Li and R.W. Yeung, “Network Information Flow”, IEEE-IT, vol. 46, pp. 1204-1216, 2000.

[3] R. Koetter and M. Médard, “An Algebraic Approach to Network Coding”, Proceedings of the 2001 IEEE International Symposium on Information Theory, pp. 104, 2001.

[4] R. Koetter and M. Médard, “Beyond Routing: An Algebraic Approach to Network Coding”, Proceedings of the 2002 IEEE Infocom, 2002.

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

[6] T. Ho, M. Médard and R. Koetter, “A coding view of network capacity, recovery and management”, Proceedings of the 2002 IEEE International Symposium on Information Theory, 2002.