An Information-Theoretic View of Network Management

Tracey Ho, *Member, IEEE*, Muriel Médard, *Senior Member, IEEE*, and Ralf Koetter, *Member, IEEE*

***Abstract*—We present an information-theoretic framework for network management for recovery from nonergodic link failures. Building on recent work in the field of network coding, we describe the input–output relations of network nodes in terms of network codes. This very general concept of network behavior as a code provides a way to quantify essential management information as that needed to switch among different codes (behaviors) for different failure scenarios. We compare two types of recovery schemes, receiver-based and network-wide, and consider two formulations for quantifying network management. The first is a centralized formulation where network behavior is described by an overall code determining the behavior of every node, and the management requirement is taken as the logarithm of the number of such codes that the network may switch among. For this formulation, we give 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 connections and an upper bound for multitransmitter multicast connections, for linear receiver-based and network-wide recovery from all single link failures. The second is a node-based formulation where the management requirement is taken as the sum over all nodes of the logarithm of the number of different behaviors for each node. We show that the minimum node-based requirement for failures of links adjacent to a single receiver is achieved with receiver-based schemes.**

***Index Terms*—Graph theory, network coding, network management, network restoration, Shannon theory.**

I. Introduction

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. Fig. 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.

To illustrate this point, we consider two of the most common means of providing network recovery for nonergodic 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 [1], [2]. Path restoration may be itself subdivided into two different types: live (dual-fed) backup and event-triggered backup. In the first case, two live flows, a primary and a backup, 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 backup capacity is not shared among connections. In the second case, event-triggered path protection, the backup 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 [3], thus improving capacity utilization for backup channels and allowing for judicious planning [4]–[12]. However, recovery involves coordination between the sender and the receiver after a failure event and requires action from nodes along the backup path.

The simplest scheme to consider is live path protection, shown in Fig. 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 Fig. 1(a).

Manuscript received July 28, 2003; revised June 2, 2004. This work was supported in part by the National Science Foundation under Grants CCR-0325496, CCR-0093349, by AFOSR under Grant PT. HoY-1362, and by the HP Wireless Center. T. Ho performed this work while working toward the Ph.D. degree at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge.

T. Ho is with Bell Laboratories, Murry Hill, NJ 07974 USA (e-mail: trace@mit.edu).

M. Médard is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: medard@mit.edu).

R. Koetter is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: koetter@csl.uiuc.edu).

Communicated by G. H. Sasaki, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2005.844062

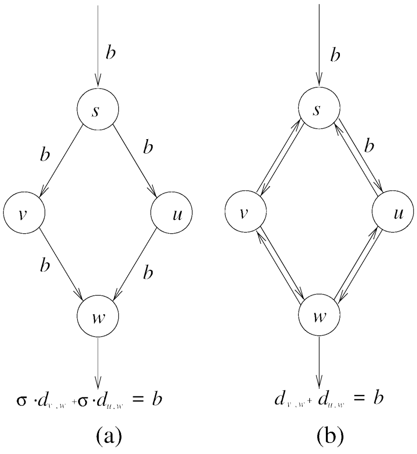

Fig. 1. Ring network illustrating path protection in (a) and link protection in (b).

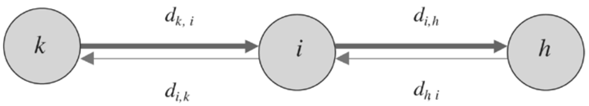

Fig. 2. Three nodes and their primary (thick) and backup (thin) links.

For failure-induced path protection, the sender knows σ. The code is similar to the one in Fig. 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 offailure. 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 Fig. 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 Fig. 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 $i = s$, for which it is the incoming signal b. For the secondary link emanating from i, the code is $d_{i,k} = d_{h,i} \cdot \sigma_{i,h} + d_{i,h} \cdot \overline{\sigma}_{i,h}$. The output at node w, as shown in Fig. 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.

Our above example 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 [13]–[16] or generalized loop-back [17]. 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.

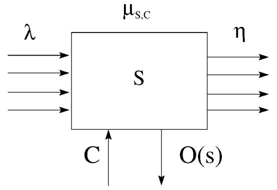

Fig. 3. Diagram of general network management problem.

We may formulate a basic general form of the network management problem as shown in the block diagram in Fig. 3. A network is modeled 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 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 paper 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 nonergodic 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.

Our approach has its roots in recent work on network coding [18], [19]. Ahlswede *et al.* [19] showed that the traditional approach of transmitting information by routing or replication is not always sufficient to achieve maximum capacity for multicast, and that this sometimes requires coding together signals from different incoming links. Koetter and Médard [20]–[22] introduced an algebraic framework for analyzing network coding, and showed that with coding, a multicast network has a linear *receiver-based* solution for all recoverable failures, defined as a solution in which only the receiver nodes react to the failure pattern, while the other nodes (interior nodes) do 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.

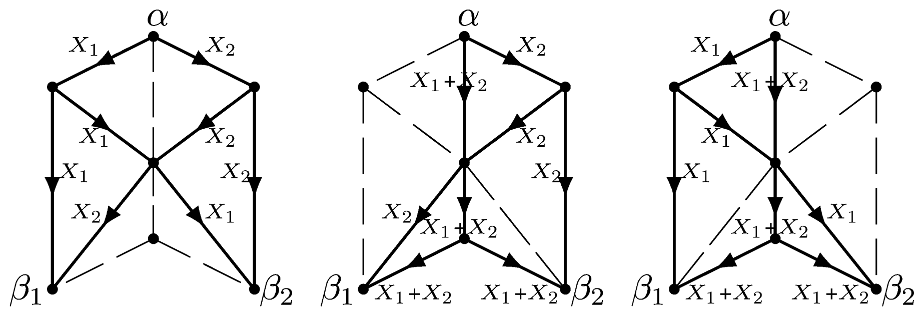

Fig. 4. An example of a receiver-based recovery scheme. Each diagram corresponds to a code valid for failure of any of the links represented by dashed lines. The only nodes that alter their input–output relations across the three codes are the receiver nodes β_1 and β_2.

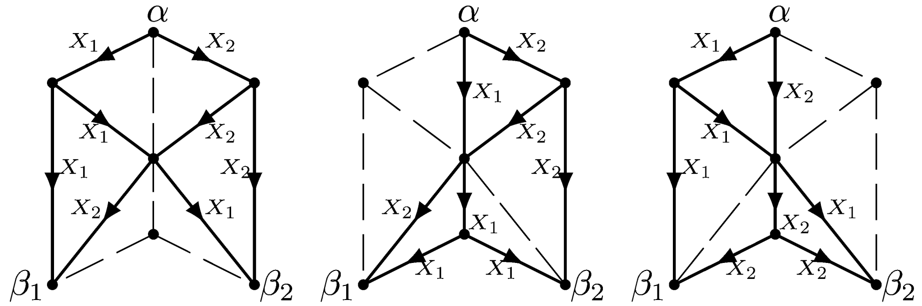

Fig. 5. An example of a network-wide recovery scheme. Each diagram gives a code which is valid for failure of any of the links represented by dashed lines.

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 involves 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 Figs. 4 and 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 Fig. 4, and a possible set of codes forming a network-wide scheme is given in Fig. 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 Fig. 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 Fig. 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 $\log(3) + 2\log(2) = 3.58$.

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 multitransmitter 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-link failures.

Parts of this work have been presented in [23], which considered the multitransmitter single-receiver case (where there is only one receiver node), [24], which considered failures of links adjacent to the receiver nodes in the multitransmitter multicast case, and [25], which considered general connections.

We present our model in Section II, state our main results in Section III, give our mathematical development, ancillary results, and proofs in Section IV, and give conclusions and directions for further work in Section V.

II. Model

Our model is based on that in [20]. We represent a network by a directed graph $\mathcal{G}$ with vertices representing nodes and ν directed edges representing links. In this paper, we consider only delay-free acyclic networks. Discrete independent random processes $X_1, \ldots, X_r$ are observable at one or more source nodes, and processes originating at different source nodes are independent. There are one or more receiver nodes, comprising a set $\mathcal{D}$. A network connection problem specifies, for each receiver node $\beta \in \mathcal{D}$, a subset $\mathcal{X}_\beta$ of source processes to be transmitted to β. A network connection problem is *feasible* if the network

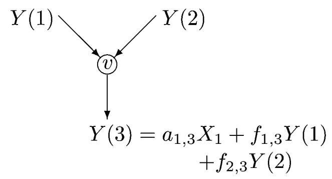

Fig. 6. Illustration of linear coding at a node.

supports the replication, concurrently at each receiver $\beta \in \mathcal{D}$, of each source process $X_i \in \mathcal{X}_\beta$, i.e., β has an output process $Z(\beta, i)$ that is a copy of X_i. A subgraph is said to support a set of connections if the connections are feasible after deletion of all links not in the subgraph. A multicast connection problem entails the transmission of the same set of source processes to each of the receiver nodes.

Edge l carries the random process $Y(l)$. Edge 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 incoming link of a receiver node a *terminal link*, and other links *interior links*.

We choose the time unit such that the capacity of each link is one bit per unit time, and the random processes X_i have a constant entropy rate of one bit per unit time. Sources of larger entropy rate can be modeled as multiple sources at the same node. Edges with larger capacities can be modeled as parallel edges, though in this paper we focus on management requirements for failure of individual unit capacity components.

The processes X_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 *et al.* [18] to be sufficient for multicast. In a linear code, 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 (refer to Fig. 6)

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

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

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

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 $B_\beta = (b_{\beta_{i,l}})$, and the $\nu \times \nu$ matrix $F = (f_{l,j})$, whose structure is constrained by the network. 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.

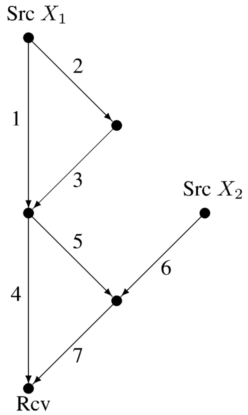

Fig. 7. Example illustrating integral links and recoverable link failures. Sources X_1 and X_2 are required to be concurrently transmitted to the receiver. Links 1, 2, and 3 are integral, and failure of any one of them is recoverable. Links 4, 6, and 7 are integral, but their failures are not recoverable.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

We also 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 IV-B, restricts the class of recovery codes that can be used. For the linear coding matrices described above, failure of a link h corresponds to setting to zero the hth column of matrices A, B, and F, and the hth 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

In this section, we outline our main results, the proofs of which are given in the following section. 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 Fig. 7.

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. □

Although a solution with static A and F matrices always exists for any recoverable set of failures in a multicast scenario [20], in such cases the receiver code B must change. On the other hand, if we allow for nonlinear processing at the receivers, in some instances this allows for unchanged network behavior over all recoverable failures.

Theorems 2–4 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 in 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 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	$\left\lceil \frac{m+1}{m-r+1} \right\rceil$

□

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 as shown in the table at bottom of the page.

b. For the **multicast case with** $d \geq 2$ **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 **nonmulticast 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 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. Fig. 8 gives a plot of how the bounds look for single-receiver networks with minimum cut size $m = 8$.

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.

Theorem 4 (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
$\begin{cases} r & \text{for } 1 < r = t_{\min} - 1 \\ 1 & \text{for } r = 1 \text{ or } r \leq t_{\min} - 2 \end{cases}$	r

□

Related work by Cai and Yeung [26] 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 5 (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. □

This does not however hold for the multireceiver case. A counterexample is shown in Fig. 9. 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.

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	$\begin{cases} r+1 & \text{for } r = 1, r = 2 = m-1 \\ r & \text{for } r = 2 \leq m-2, r = 3, r = m-1 \geq 3 \\ r-1 & \text{for } 4 \leq r \leq m-2 \end{cases}$

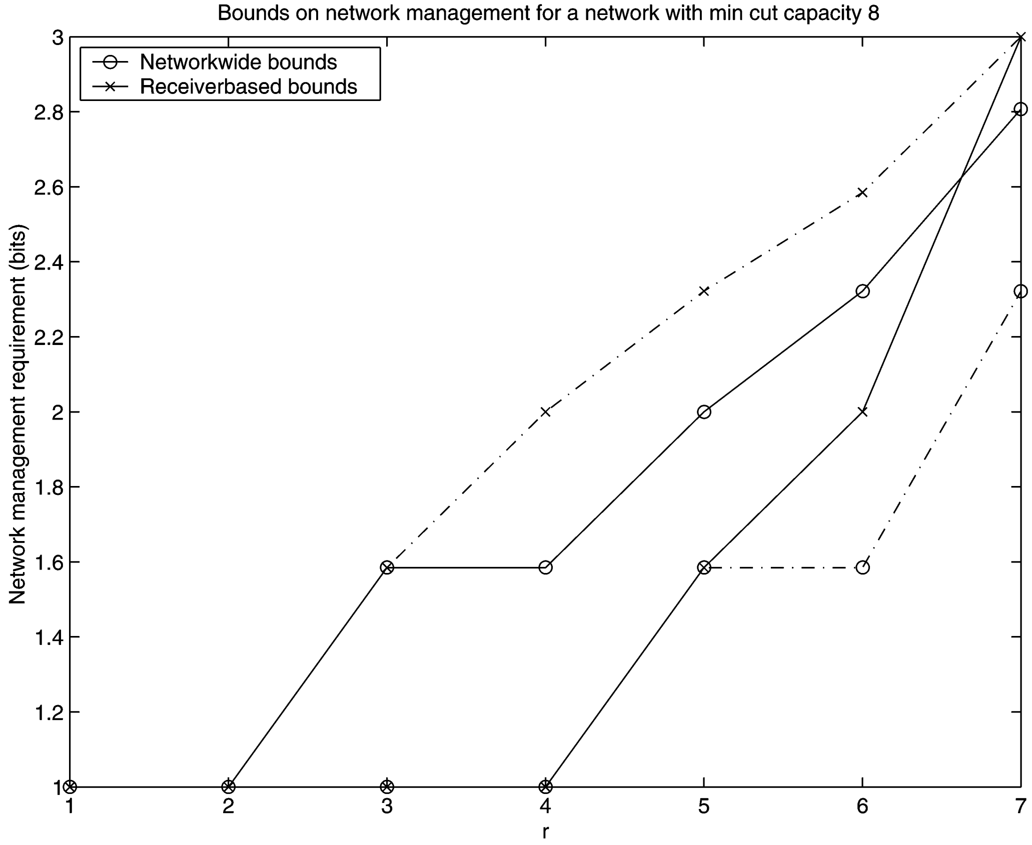

Fig. 8. Plot of tight upper and lower bounds for centralized network management in single-receiver networks with fixed minimum cut size $m = 8$ and arbitrary numbers of links. The parameter r denotes the number of source processes transmitted in the network.

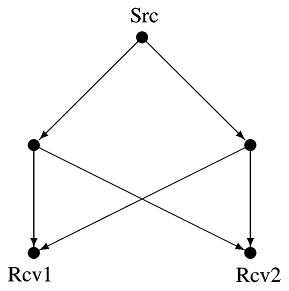

Fig. 9. Counterexample showing that Theorem 5 does not hold for the multireceiver case.

IV. Detailed Development, Ancillary Results, and Proofs

A. 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 $\underline{Z}$ [20]. $(I - F)$ is always invertible since F is upper-triangular for acyclic networks. 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. $G = I + F + F^2 + \cdots$ sums the gains along all paths between each pair of links, and equals $(I-F)^{-1}$, since matrix F is nilpotent for acyclic networks. 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 paper.

- $M(i, j)$ 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.

Fig. 10 gives an example illustrating the structure of the transfer matrices for a single receiver β and n source nodes α_j, each with r_{α_j} source processes.

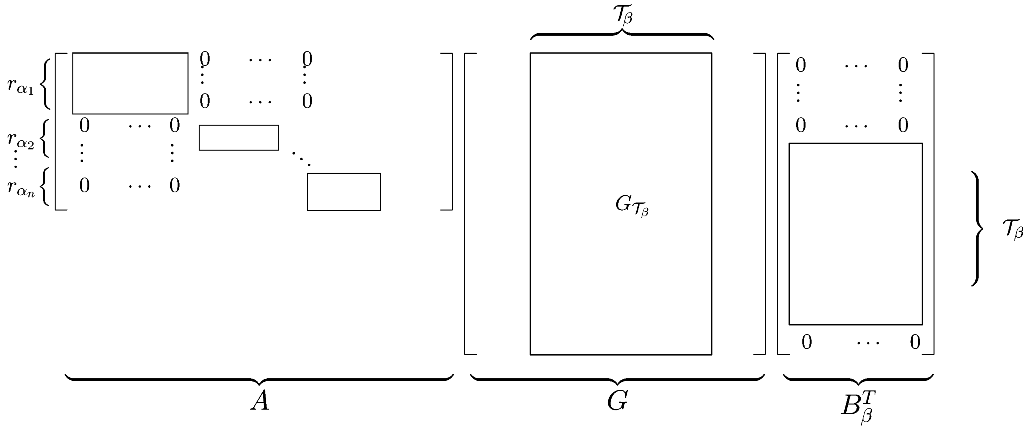

Fig. 10. An example illustrating the structure of transfer matrices for a single receiver β and n source nodes α_j, each with r_{α_j} source processes.

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 ith 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

$$G^h = I^h + F^h + (F^h)^2 + \cdots = I^h(I - FI^h)^{-1} = (I - I^hF)^{-1}I^h.$$

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.

B. 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 1 (Codes Covering Multiple Scenarios):

1. If code (A, G, B) covers the no-failure scenario and failure of link h, then
$$\underline{c}_h \sum_{j \in \mathcal{T}_\beta^h} G(h, j)\underline{b}_j^T = \mathbf{0}, \qquad \forall \, \beta \in \mathcal{D}$$
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
   a)
$$\underline{c}_h \sum\nolimits_{j \in \mathcal{T}_\beta^h} G(h, j)\underline{b}_j^T = \mathbf{0}$$
   and
$$\underline{c}_k \sum\nolimits_{j \in \mathcal{T}_\beta^k} G(k, j)\underline{b}_j^T = \mathbf{0}$$
   or
   b)
$$\gamma_{h,k} \sum\nolimits_{j \in \mathcal{T}_\beta^h} G(h, j)\underline{b}_j^T = \sum\nolimits_{j \in \mathcal{T}_\beta^k} G(k, j)\underline{b}_j^T \neq \mathbf{0}$$
   and
$$c_h = \gamma_{h,k}\underline{c}_k \neq \mathbf{0}$$
   where
$$\gamma_{h,k} \in \mathbb{F}_{2^u}.$$

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

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

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

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

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

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

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

$$\sum_{j \in \mathcal{T}^h_\beta} G(h, j)\underline{b}^T_j \quad \text{and} \quad \sum_{j \in \mathcal{T}^k_\beta} G(k, j)\underline{b}^T_j$$

which, respectively, span the row spaces of the left- and right-hand side expressions satisfy

$$\gamma_{h,k} \sum_{j \in \mathcal{T}^h_\beta} G(h, j)\underline{b}^T_j = \sum_{j \in \mathcal{T}^k_\beta} G(k, j)\underline{b}^T_j. \qquad \blacksquare$$

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 $AG^h B^T_\beta$ is affected by the value on link h, i.e., $\underline{c}_h \sum_{j \in \mathcal{T}^h_\beta} G(h, j)\underline{b}^T_{\beta_j} \neq \mathbf{0}$. A code is *active* in a link h if it is active in h for some receiver β. Otherwise, the code is *nonactive* in h. For a code which is nonactive in a link h, the value on h could be set to zero (by upstream links ceasing to transmit on the link), canceled out, or disregarded by the receivers.

By Part 1 of Lemma 1, a code which covers the no-failure scenario as well as one or more single-link failures must be nonactive in those links. By Part 2 of Lemma 1, a code which covers failures of two or more single links is, for each receiver, either nonactive 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 *nonactive* otherwise. If signals on failed links are undetermined, then consideration must be restricted to nonactive codes.

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)
   $$\underline{c}_h \underline{b}^T_h = \mathbf{0} \quad \text{and} \quad \underline{c}_k \underline{b}^T_k = \mathbf{0}$$
   or

   b) h and k are terminal links of the same receiver
   $$\gamma_{h,k}\underline{b}^T_h = \underline{b}^T_k \neq \mathbf{0} \quad \text{and} \quad \underline{c}_h = \gamma_{h,k}\underline{c}_k \neq \mathbf{0}$$
   where $\gamma_{h,k} \in \mathbb{F}_{2^u}$. □

Proof of Theorem 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 nonactive in k. However, a code which is nonactive 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 $\mathcal{H}$. ■

C. Bounds on Linear Network Management Requirement

1) 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*. $q_{i,j}$ and $d_{l,j}$ (which correspond exactly to $a_{i,j}$ 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 $\mathcal{J}$. 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 2: 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 $\mathcal{M}$;

[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.

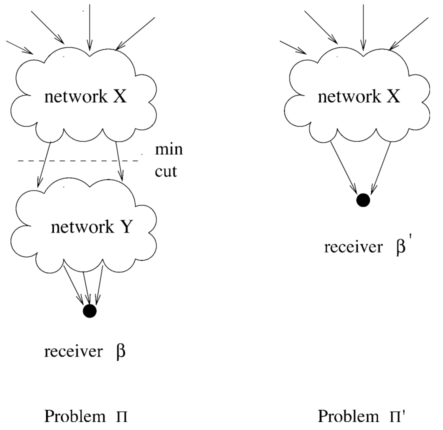

Fig. 11. Example illustrating the definition of a related connection problem Π' from an original problem Π.

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

An example illustrating this is given in Fig. 11.

Corollary 2: 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 3: 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

$$AG^hB^T = QJ_{\mathcal{M}}^{h}L^T = I, \qquad \forall\, h \in \mathcal{M}_1.$$

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 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 single-terminal link failures is

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

Proof: For $r = 1$, $\lceil \frac{t}{t-r} \rceil = 2$. Just two codes are needed as only one of the links needs to be active in each code. For $t = r + 1$, $\lceil \frac{t}{t-r} \rceil = r + 1$. 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 $\{\underline{w}_k | k = 1, \ldots, t - r, k \neq i', j'\}$ by another code, and the links corresponding to $\{\underline{v}_k | k = 1, \ldots, r, k \neq i, j\}$ by a separate code each. ■

Lemma 5: 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 nonactive 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 $(\leq n)$ nonactive codes are required, one nonactive in the primary links and the other nonactive in the rest. Otherwise, there is some nonactive code in the set, or some active code covering only one terminal link failure, which can be replaced by a corresponding nonactive code covering that link. The links covered by this nonactive code can be covered together with the primary links of the active codes, with a single nonactive code. The secondary links of the active codes can be covered by a separate nonactive code. This forms a set of at most n nonactive codes covering the same terminal link failures as the original set. ■

Corollary 3: For receiver-based recovery, the minimum number of codes for terminal link failures can be achieved with nonactive 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 table at the top of the following page. These bounds are the same in the case where only nonactive codes are used.

Proof: It follows from Lemma 2 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

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

links $h \in \mathcal{M}_1$ in the original problem (Lemma 3). We can thus apply the upper bound from Lemma 4 with m in place of t.

For the lower bound, from Lemma 1, a single nonactive code in a valid receiver-based scheme can cover at most $m - r$ of the links in $\mathcal{M}$. By Corollary 3, restricting consideration to nonactive 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 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 $\lceil \frac{m+1}{m-r+1} \rceil$.

Proof: It follows from Lemma 1 that a single nonactive 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: From Corollary 2, 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' [22]. 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 $QJ_{\mathcal{M}}^{h'} = AG_{\mathcal{M}}^{h}$, which has full rank. For failure of any link $k \notin P$, $\underline{c}_h$ is available on h_i via P, so

$$\operatorname{rank}\left(QJ_{\mathcal{M}}^{k}\right) = \operatorname{rank}\left(AG_{\mathcal{M}}^{k}\right) = r.$$

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 $AG_{\mathcal{L}}^{h}$ 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 $AG_{\mathcal{M}}^{h}$ has full rank and $m > r$, $AG_{\mathcal{M}}^{h}$ 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 $AG_{\mathcal{M}}^{h_i}$ 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 $AG_{\mathcal{L}}^{h}$ 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. Again, 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' 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 in this family of codes, corresponding to different choices of values for variables d_{l_1,l_2}. 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 [22].

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 $QJ'^{h'}_{\mathcal{L}} = AG_{\mathcal{L}}^{h}$ by the assumption of this case. Hence, failure of h' is covered by (Q, J').

Next, consider some link $k \notin R$. If $AG_{\mathcal{M}}^{\{h,k\}}$ 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.

$$\left[\begin{array}{cccccccccccccccc}
x & 0 & \cdots & 0 & x & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & x & \cdots & 0 & x & 0 & \cdots & \cdots & & & & & & & & \vdots \\
\vdots & & \ddots & \vdots & & & & & & & & & & & & \vdots \\
\vdots & & & x & x & & & & & & & & & & & \vdots \\
\vdots & & & & & x & 0 & \cdots & 0 & x & & & & & & \vdots \\
\vdots & & & & & & x & \cdots & 0 & x & & & & & & \vdots \\
\vdots & & & & & & & \ddots & \vdots & & & & & & & \vdots \\
\vdots & & & & & & & & x & x & & & & & & \vdots \\
\vdots & & & & & & & & & & \ddots & & & & & \vdots \\
\vdots & & & & & & & & & & & x & 0 & \cdots & 0 & x \\
\vdots & & & & & & & & & & & & x & \cdots & 0 & x \\
\vdots & & & & & & & & & & & & & \ddots & & \vdots \\
0 & & & & & & & & & & & & & & x & x
\end{array}\right]
\begin{array}{l}
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \left\lceil\frac{m}{m-r}\right\rceil - 1 \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \left\lceil\frac{m}{m-r}\right\rceil - 1 \\
\\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} m-(m-r-1)\left\lceil\frac{m}{m-r}\right\rceil - 1
\end{array}$$

Fig. 12. An example of an $AG_{\mathcal{M}}$ matrix in a receiver-based code that achieves the lower bound of $\lceil\frac{m}{m-r}\rceil$ codes, with $m-(m-r-1)\lceil\frac{m}{m-r}\rceil \geq 2$.

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

Now $AG_{\mathcal{M}}^{\{h,k,h_j\}}$ cannot have rank $\leq r-2$ since this would mean that $AG_{\mathcal{M}}^{\{h,h_j\}}$ has at most rank $r-1$, and the full rank assumption on $AG_{\mathcal{L}}^{h}$, whose column space is contained in the column space of $AG_{\mathcal{M}}^{\{h,h_j\}}$, would be contradicted. Thus, $AG_{\mathcal{M}}^{\{h,k,h_j\}}$ 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 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 (Lemma 8). A nonactive 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 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 Fig. 12.

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 8, 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 ith 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 $\underline{e}_i$ for some process i 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 case b, 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 $r-1$ nonactive codes. This is because the remaining $r-1$ links in

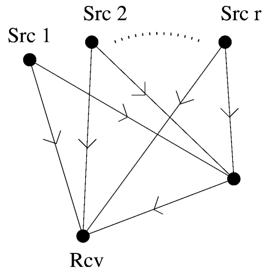

Fig. 13. An example network in which $r = m - 1$, which achieves the linear receiver-based upper bound of $r + 1$ codes and the linear network-wide and nonlinear receiver-based upper bounds of r codes.

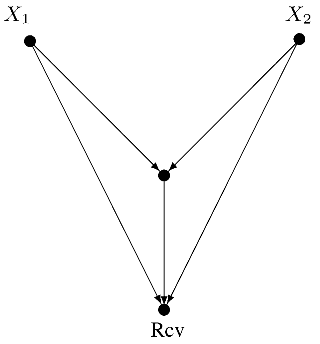

Fig. 14. An example network in which $r = 2 = m - 1$, which achieves the linear network-wide upper bound of three codes.

$\mathcal{M}$, and their associated subgraphs, can be covered by nonactive 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 nonactive codes.

An example in which $r = m - 1$, and r network-wide codes are needed is given in Fig. 13. This is not the case for $r = 2 = m-1$, for which an example requiring three network-wide codes is given in Fig. 14.

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 nonactive code, and that $r + 2 - 3 = r - 1$ nonactive codes suffice to cover all columns.

Let the columns in this set be $\underline{u}_1, \dots, \underline{u}_r, \underline{w}_1, \underline{w}_2$, where $\underline{u}_1, \dots, \underline{u}_r$ form a basis, and let the remaining columns in $AG_{\mathcal{M}}$ be $\underline{w}_3, \dots, \underline{w}_{m-r}$. Expressing each $\underline{w}_i$ as a linear combination $\underline{w}_i = \lambda_{i,1}\underline{u}_1 + \dots + \lambda_{i,r}\underline{u}_r$, the pairwise independence of columns in the set implies that for $i = 1$ and $i = 2$, at least two of $\lambda_{i,1}, \dots, \lambda_{i,r}$ 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 without loss of generality that $\lambda_{1,k}, \lambda_{2,l} \neq 0$. By the assumption of recoverability, at least one of $\lambda_{1,j}, \dots \lambda_{m-r,j}$ 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

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

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 nonactive code, along with links corresponding to any columns $\underline{w}_3, \dots, \underline{w}_{m-r}$.

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

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

is a full-rank set, as are

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

where l' is distinct from k', k, l; and

$$m' \in \{1, 3, 4, \dots, |\mathcal{M}| - r\}, \qquad \lambda_{m',l'} \neq 0.$$

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 $\underline{w}_3, \dots, \underline{w}_{m-r}$. 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 $\underline{w}_3, \dots, \underline{w}_{m-r}$. 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 $2 \leq \tilde{r} + 1 \leq 3$ nonactive codes if $\tilde{r} = 1, 2$, and by $\tilde{r}$ nonactive 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 $\tilde{r} - 1 \geq 2$ nonactive codes suffice to cover the $\tilde{\nu}$ links and their associated subgraphs. Any two nonactive codes covering these links can also cover the remaining links corresponding to the

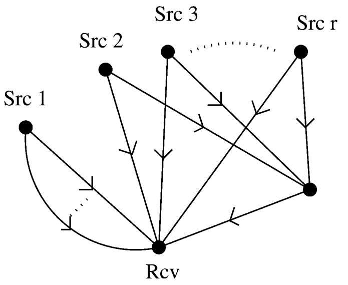

Fig. 15. An example network which achieves the receiver-based upper bound of r, the network-wide upper bounds of r codes for $r = 3$, and $r - 1$ codes for $4 \leq r \leq m - 2$.

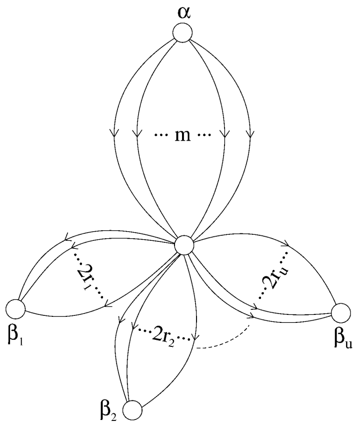

Fig. 16. An example network which achieves the general case lower bounds of Theorem 2 with equality, where r_i is the number of processes received by receiver β_i.

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 three.

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 Fig. 15 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 $4 \leq r \leq m - 2$. ■

2) 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 $\lceil \frac{m}{m-r} \rceil$ for receiver-based recovery and $\lceil \frac{m+1}{m-r+1} \rceil$ 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 Fig. 16, 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 $\lceil \frac{m}{m-r} \rceil$ nonactive 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 $\lceil \frac{m+1}{m-r+1} \rceil$ 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 $\lceil \frac{m+1}{m-r+1} \rceil$ codes that includes

$$\min\left(\left\lceil \frac{m+1}{m-r+1} \right\rceil, y + 1 \right) \geq 2$$

nonactive codes, which can at the same time cover all the terminal links.

For the case where $\lceil \frac{m+1}{m-r+1} \rceil$ is an integer, however, covering links on the minimum cut with exactly $\lceil \frac{m+1}{m-r+1} \rceil$ codes would allow for only one nonactive code (Lemma 7), so this bound is not attained with equality for two or more receiver nodes. ■

3) 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 3 and 8, 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 toward 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 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 3a.

Proof of Theorem 3b: We first analyze the two-receiver case, considering three cases.

Case 1: $2 \leq r \leq m_{\beta_x} - 2$ for both receivers β_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 3a), 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 $\mathcal{B}_1^k, k \le r$, each paired with a distinct tree $\mathcal{G}_1^{\gamma_k}$, such that the 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 $s_{\beta_2} \le r$, 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 $\mathcal{B}_2^l, l \le r$, each paired with a distinct tree $\mathcal{G}_2^{\gamma_l}$, such that the 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 outputs a possibly modified set of paths. 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. 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 Fig. 17. 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.

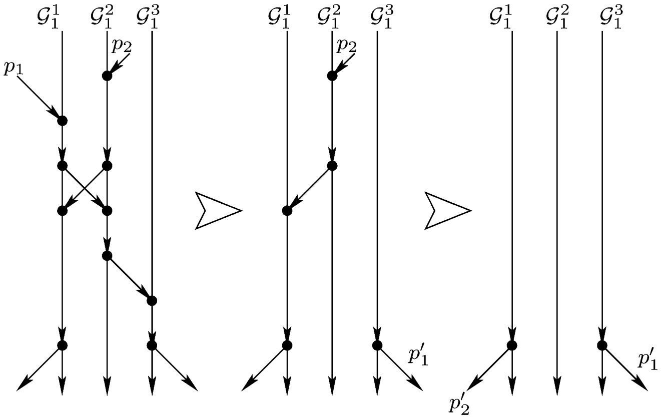

Fig. 17. An illustration of the path-shortening procedure. In the first step, path P_1 is shortened to form P_1' by removing the portion of P_1 upstream of its intersection with tree $\mathcal{G}_1^3$. In the second step, path P_2 is shortened to form P_2' by removing the portion of P_1 upstream of its intersection with tree $\mathcal{G}_1^1$.

This leads to the following property.

Property: 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 nth 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 signal vectors of the leading intersections of any r trees form a basis set. There exists also a code in which the signal 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 signal 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 suffices 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 does not suffice for transmission to β_2. Then the set $\overline{\mathcal{V}}_n = \mathcal{P}_n - \mathcal{V}_n$ is nonempty, 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 $\underline{w} \in \mathrm{span}(\mathcal{W}_n)$ as a linear combination of vectors in $\mathcal{R}(\mathcal{U}_n)$ includes at least one vector $\underline{v}_k, 1 \le k \le |\mathcal{W}_n|$. 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 nonempty 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

$$|\mathcal{T}_n| \ge r + 1 - |\overline{\mathcal{V}}_{n-1}|.$$

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 nonempty.

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 $|\overline{\mathcal{V}}| \le r$, and the set $\mathcal{W}$ for each run of the procedure ending in Case 3b must be nonempty, 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, $3 \le x \le d$, can be grouped into $s_{\beta_x} \le r + 1$ link-disjoint forests (Theorem 3a), 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 $\le (r^2 + 2)(r + 1)$ subsets may, in turn, contain links that are part of $\le r + 1$ 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 multireceiver all link failures case. For the two-receiver case, an example in which $(r + 1)(r + 2)/2$ codes are needed is given in Fig. 18. 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 nonmulticast 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 nonmulticast case, the number of codes for terminal link failures can grow linearly in the number of receivers.

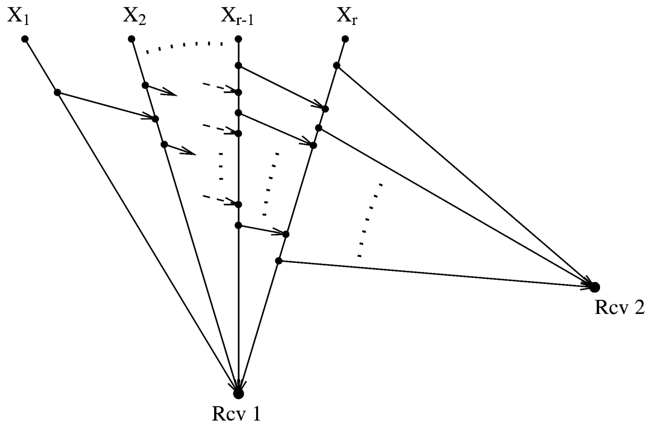

Fig. 18. An example multicast problem in which $(r+1)(r+2)/2$ codes are needed for all link failures.

Proof of Theorem 3c: We will use nonactive 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 nonmulticast problem. From Theorem 1, at least two 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. Fig. 19 gives an example which comes close to this bound, requiring

$$\sum_{\beta t_\beta \leq r} (t_\beta - 2) + \sum_{\beta t_\beta \geq r+1} (r-1)$$

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 $h_{i,i+1}$ 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 $h_{i,i+1}$ to carry the other process only, necessitating separate codes.

D. Nonlinear Receiver-Based Recovery

Proof of Theorem 4: 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 failure 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 two positions which are both nonzero in at least one of the codewords.

First we consider the lower bound. 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 two 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 one 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 two 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 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 Fig. 20. 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 Fig. 13.

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

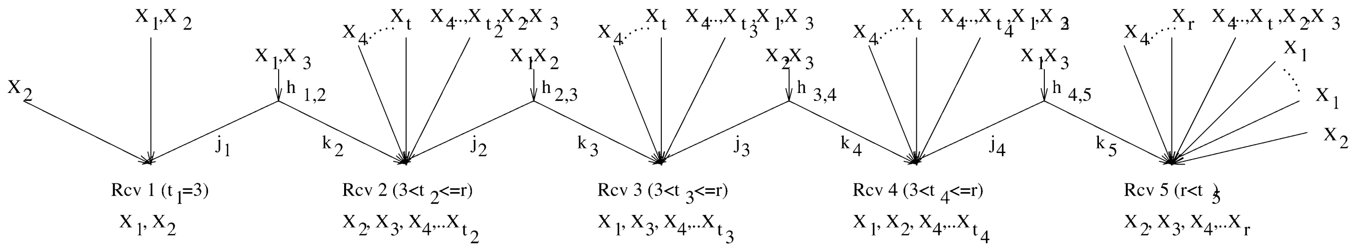

Fig. 19. An example network in which $\sum_{\beta t_\beta \leq r}(t_\beta - 2) + \sum_{\beta t_\beta \geq r+1}(r-1)$ codes are needed.

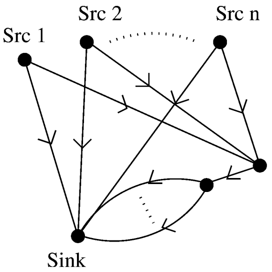

Fig. 20. An example network in which $2 \leq r \leq t-2$, which achieves the nonlinear receiver-based upper bound of r codes.

in a multiple-receiver problem such that each receiver requires n_β nonlinear codes requires $\max_\beta n_\beta$ codes in total.

E. Node-Based Management Requirement

To prove Theorem 5, we first establish the following lemmas.

Lemma 9: In a given network, for any set of nonactive 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 nonactive 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 4: 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 1). Only one of these columns is needed to form a basis, so a single nonactive code can cover all but one of these links, and another nonactive code can cover the remaining link. The result follows from applying Lemma 9. ■

Lemma 10: 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 1), there is at least one nonactive code. If codes

$$\{(A_1, G_1, B), (A_2, G_2, B), \ldots, (A_n, G_n, B)\}$$

are all nonactive, there is a set of n codes with common (A, G) that cover the same terminal link failures (Lemma 9).

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 nonactive 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 nonactive 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 nonactive code, and its primary link can be covered by a new nonactive code. A set of n nonactive codes covering the same failures as the original set can thus be constructed. By Lemma 9, there exists a set of n receiver-based codes covering the same failures. ■

Proof of Theorem 5: 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

$$\sum_{i=1}^{x} \log_2 m_i + \log_2 n = \log_2 \left(\Pi_{i=1}^{x} m_i\right) n \geq \log_2 mn$$

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 10, 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

$$\{(A, G, B_1), (A, G, B_2), \dots, (A, G, B_{n_1})\}$$

has a common AG matrix, there is a corresponding set of $\leq n_1$ nonactive codes covering the same terminal links (Lemma 5). Each of the remaining codes can be covered by one or two nonactive codes (Corollary 4). Replacing active codes by nonactive codes in this way, the maximum resulting number of nonactive 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 an AG matrix that is the same as for $n-1$ other codes, and mn nonactive codes suffice. If there are $k \geq 1$ AG matrices that are not common across two or more codes, then the number of nonactive codes needed is at most

$$(m-k)n + 2k = mn - k(n-2) \leq mn, \qquad \text{for } n \geq 2.$$

Thus, there exists a complete set of $\leq mn$ receiver-based codes (Lemma 9). ∎

V. Conclusion 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. We have compared the management requirements for network-wide and receiver-based recovery, and have provided bounds on network management 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.

Several areas of further research result from this work. One such area 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 paper may be useful.

References

[1] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks, part I—Protection," in *Proc. IEEE INFOCOM*, New York, Mar. 21–25, 1999, pp. 744–751.

[2] ——, "Survivable WDM mesh networks, Part II—Restoration," in *Proc. IEEE Int. Conf. Communications*, Vancouver, BC, Canada, Jun. 6–10, 1999, pp. 2023–2030.

[3] C. S. Wu, S. W. Lee, and Y. T. Hou, "Backup vp preplanning strategies for survivable multicast ATM," in *Proc. IEEE Int. Conf. Communications*, Montreal, QC, Canada, Jun. 8–12, 1997, pp. 267–271.

[4] M. Barezzani, E. Pedrinelli, and M. Gerla, "Protection planning in transmission networks," in *Proc. IEEE Int. Conf. Communications*, vol. 2, Chicago, IL, Jun. 1992, pp. 316.4.1–316.4.5.

[5] M. Herzberg, S. J. Bye, and A. Utano, "The hop-limit approach for spare-capacity assignment in survivable networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 6, pp. 775–784, Dec. 1995.

[6] W. D. Grover, T. D. Bilodeau, and B. D. Venables, "Near Optimal Synthesis of a Mesh-Restorable Network," Telecom Canada, Rep. CR-90-19-01, Feb. 1991.

[7] M. Herzberg and S. J. Bye, "An optimal spare-capacity assignment model for survivable networks with hop limits," in *Proc. IEEE GLOBECOM*, vol. 3, San Francisco, CA, Nov./Dec. 1994, pp. 1601–1606.

[8] A. Gersht, S. Kheradpir, and A. Shulman, "Dynamic bandwidth-allocation and path-restoration in SONET self-healing networks," *IEEE Trans. Reliab.*, vol. 45, no. 2, pp. 321–331, Jun. 1996.

[9] H. Sakauchi, Y. Nishimura, and S. Hasegawa, "A self-healing network with an economical spare-channel assignment," in *Proc. IEEE GLOBECOM*, vol. 1, San Diego, CA, Dec. 1990, pp. 403.1.1–403.1.6.

[10] B. D. Venables, W. D. Grover, and M. H. MacGregor, "Two strategies for spare capacity placement in mesh restorable networks," in *Proc. IEEE Int. Conf. Communications*, vol. 1, Geneva, Switzerland, May 1993, pp. 267–271.

[11] T. Frisanco, "Optimal spare capacity design for various protection switching methods in ATM networks," in *Proc. IEEE Int. Conf. Communications*, vol. 1, Montreal, QC, Canada, Jun. 8–12, 1997, pp. 293–298.

[12] N. H. M. Nizam, G. K. Hunter, and D. G. Smith, "A dynamic reconfiguring tool for improving multiwavelength transport network robustness," in *Proc. IEEE Int. Conf. Communications*, vol. 1, Montreal, QC, Canada, Jun. 8–12, 1997, pp. 246–250.

[13] O. J. Wasem, "An algorithm for designing rings for survivable fiber networks," *IEEE Trans. Reliab.*, vol. 40, no. 4, pp. 428–432, Oct. 1991.

[14] W. D. Grover, "Case studies of survivable ring, mesh and mesh-arc hybrid networks," in *Proc. IEEE GLOBECOM*, Orlando, FL, Dec. 6–9, 1992, pp. 633–638.

[15] G. Ellinas, T. E. Stern, and A. Hailemarian, "Link failure restoration in optical networks with arbitrary mesh topologies and bi-directional links," *IEEE J. Sel. Areas Commun.*, to be published.

[16] G. Ellinas and T. E. Stern, "Automatic protection switching for link failures in optical networks with bi-directional links," in *Proc. IEEE GLOBECOM*, London, U.K., Nov. 18–22, 1996.

[17] M. Médard, R. A. Barry, S. G. Finn, W. He, and S. S. Lumetta, "Generalized loop-back recovery in optical mesh networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 1, pp. 153–164, Feb. 2002.

[18] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[19] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1204–1216, Jul. 2000.

[20] R. Koetter and M. Médard, "An algebraic approach to network coding," in *Proc. IEEE Int. Symp. Information Theory*, Washington, DC, Jun. 2001, p. 104.

[21] ——, "Beyond routing: An algebraic approach to network coding," in *Proc. IEEE INFOCOM*, New York, June 23–27, 2002.

[22] ——, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[23] T. Ho, M. Médard, and R. Koetter, "A coding view of network recovery and management for single-receiver communications," in *Proc. 2002 Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002.

[24] ——, "A coding view of network capacity, recovery and management," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 137.

[25] ——, "An information theoretic view of network management," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 2003.

[26] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. IEEE Information Theory Workshop*, Bangalore, India, Oct. 20–25, 2002, pp. 119–122.