Allocation of Network Error Correction Flow to Combat Byzantine Attacks

Zhiqing Xiao, Yunzhou Li, *Member, IEEE*, Ming Zhao, *Member, IEEE*, Xibin Xu, *Member, IEEE*, and Jing Wang, *Member, IEEE*

***Abstract*—This paper studies the allocation of information flows in noiseless, memoryless communication networks in the presence of omniscient Byzantine adversary. In such networks, adversary may maliciously modify some edge-flows, and legitimate users should resort to network error correction strategies to transmit data reliably. Unlike prior papers, which focused on the capacities of the networks, we consider the expense of resources used by the flow. Hereby, this paper uses an optimization problem to define the concept of minimum cost network error correction flows. We provide a necessary and sufficient condition of feasibility of the allocation problem, and derive a cut-set outer bound on the feasible region. Using this cut-set bound, we can find the minimum cost network error correction flow in some instances. Moreover, we also consider the relationship between incoming edge-flows and outgoing edge-flows of a vertex. As for the directed acyclic graphs, we propose an algorithm to allocate the network error correction flow. This algorithm is with polynomial time complexity, and proves to be optimal when recoding at intermediate nodes is forbidden. Additionally, in order to justify the necessity of recoding at intermediate nodes, we analyze the benefit of intermediate recoding. On the one hand, we construct a series of instances to prove that intermediate recoding can bring enormous benefits in some networks. On the other hand, numerical analysis shows that the benefit is modest in small random graphs.**

***Index Terms*—Adversarial errors, Byzantine adversary, network error correction, cut-set bound, resource allocation, minimum cost.**

I. Introduction

NETWORK-LEVEL redundancy can be used to combat attacks on malicious nodes and links in communication networks with potential adversaries. Similar to the usage of link-level error correction codes to rectify errors within point-to-point transmissions, network error correction codes have been proposed to combat the link adversarial errors [1]–[5] and node adversarial errors [6]–[8] in networks.

Manuscript received October 8, 2014; revised April 3, 2015; accepted May 21, 2015. Date of publication June 1, 2015; date of current version July 13, 2015. This work was supported by the National Basic Research Program of China (973 Program) under Grant 2013CB329002, the National High Technology Research and Development Program of China (863 Program) under Grant 2014AA01A703, the National S&T Major Project under Grant 2013ZX03004007, and the Program for NCET in University under Grant NCET-13-0321. The associate editor coordinating the review of this paper and approving it for publication was A. Ramamoorthy.

Z. Xiao is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: xzq.xiaozhiqing@gmail.com).

Y. Li, M. Zhao, X. Xu, and J. Wang are with Research Institute of Information Technology, Tsinghua University, Beijing 100084, China (e-mail: liyunzhou@tsinghua.edu.cn; zhaoming@tsinghua.edu.cn; xuxb@tsinghua.edu.cn; wangj@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TCOMM.2015.2438811

Before network coding was proposed, one important way to defeat adversarial errors was to find several disjoint paths to provide diversity [9]. When the adversary intercepts and forges some edge-flows maliciously, the edge-flows in reliable paths can help correct the errors.

In [10], network coding was first used to correct errors. Byzantine attacks on random linear network coding were considered in [11] and [12]. Some coding bounds of network error correction codes were derived in [10], [13], and [14]. In the cases that the flow on each link is equal and the adversary can seize any z edges in the network, construction algorithms of linear network error correction codes to attain the refined Singleton bound were proposed in [14]–[16]. These results showed that, for the network error correction flow with equal flow on each edge, its capacity, defined as the supremum of all achievable message rates, is fully determined by the minimum cardinality of cuts, the number of adversarial links, and the quantity of the equal edge flow.

The network error correction flow with unequal flow on every link was first considered in [3]. In [3], the adversary can attack arbitrary z links among a network. It derived the capacity of the network error correction flow in two-node networks, which provided the tightest upper bound among all bounds that depend only on cuts. It further took the connection relationship among edges within cuts into consideration, and provided a bound that is sometimes tighter than the aforementioned cut-set bound. It also showed that this bound can be attained by a strategy called "Guess and Forward" in some four-node acyclic networks and some zigzag networks. After that, the relationship between the feedback edge-flow and the overall capacity in four-node acyclic networks was further studied in [17]. To guarantee the message transmission at a given rate in four-node acyclic networks, some bounds on the feasible range of the feedback edge-flow were obtained in [3, Section V-B] and [17].

Previous studies focused on characterizing the capacity of the network error correction flow. Unlike prior papers, we focus on the resource usage of network error correction flows. For a given network topology, a given message rate to attain, and all possible attacked link combinations, there may exist many different ways to allocate network error correction flow. Among those options, some network error correction flows are better since they use less network resources than others. Therefore, we need to study the minimum cost allocation of network error correction flows.

Example 1: In Fig. 1 (a) and (b), the flow on every edge is identical (denoted by f_0) and the adversary can attack arbitrary one link. The capacities in both networks are $2f_0$ [14]. However,

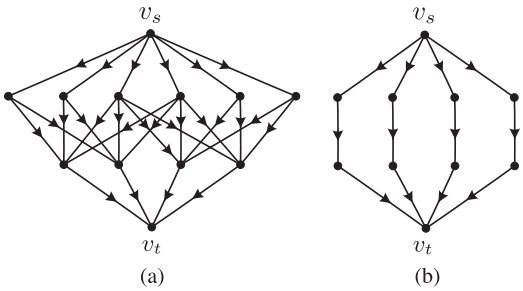


Fig. 1. Two network error correction flows with equal edge-flow on every link. (a) a network error correction flow on $6+16+4=26$ edges; (b) a network error correction flow on $4+4+4=12$ edges.

the flow in Fig. 1(a) uses more network resources and aggravates more burden than Fig. 1(b). Therefore, the flow in Fig. 1(b) is a better choice.

Cost-based network coding was considered in [18]–[20]. All these works first formulated their problems with a general cost function, but started the analysis with a very simple cost function. In [18] and [19], the cost functions are the inner product of the edge-rate and the cost per unit edge-rate. In [20], the cost is the edge-rate itself. Actually, such linear cost functions are the simplest and the most easily to be thought of, and suffice to provide intuitions for more complex cost functions.

Our previous work [21] studied the allocation of network error correction flow on disjoint paths with the linear cost functions. In [21], we proposed an algorithm to allocate the flow in the networks that consist of disjoint paths only. This algorithm proved to be optimal, and the time complexity is $O(|\mathcal{P}|\log|\mathcal{P}|)$, where $|\mathcal{P}|$ is the number of paths in the network.

In contrast, this paper allows the recoding at intermediate nodes. In fact, for a network where the capacity of each edge is infinite, if there exists a flow that requires intermediate recoding to attain a predefined rate, there must exist another flow without recoding at intermediate nodes such that it can achieve the same rate in the same network. That is, whether intermediate nodes do recoding or not does not affect the feasibility of resource allocation problem. Moreover, it can be noticed that some optimal flows do not need to recode at intermediate nodes (for example, the case of Example 1). Therefore, the benefit brought by recoding should be evaluated to justify the necessity of intermediate recoding.

Our contributions are as follows:

1) We formulate the problem of allocating network error correction flow, and discuss the feasibility of this problem. We introduce the Cost function into the system, and model the network error correction flow allocation problem as an optimization problem. We further investigate the shape of the feasible flow region, and find that the feasible region is unbounded and convex. In this sense, the feasible flow region is merely determined by its supporting hyperplanes.
2) We use a cut-based approach to analyze the feasible region. On the one hand, we derive a cut-based necessary and sufficient condition for the feasible region to be nonempty, which indicates that the feasibility of the allocation problem is merely decided by the network topology and the adversaries. On the other hand, we derive a cut-set outer bound on the feasible flow region and discuss its tightness. By proving that the cut-set outer bound is tight in a kind of two-node networks, we show that our cut-set outer bound is the tightest bound among all cut-based outer bounds on feasible region. Additionally, we also provide a counterexample to show that the cut-set outer bound is not tight in general.
3) We use the aforementioned cut-set outer bound to relax the original flow allocation problem into a linear programming, and try to find the minimum cost network error correction flow. Examples show that, although the cut-set outer bound is not tight in general, this relaxation suffices to obtain the minimum cost network error correction flow in many cases. We further discuss the relationship between the tightness of this relaxed problem and the tightness of the cut-set outer bound. Additionally, we use the idea of informational dominance to extend the classical flow conservation law for the minimum cost network error correction flow, which sometimes provides a tighter bound.
4) We propose a route-based flow allocation algorithm in directed acyclic network. The result of this algorithm is optimal among all flow allocation strategies that forbid recoding at intermediate nodes. The time complexity of this algorithm is $O\left(|\mathcal{V}|^2|\mathcal{E}|\right)$, where $|\mathcal{V}|$ is the number of vertices in the network and $|\mathcal{E}|$ is the number of edges in the network.
5) We investigate the benefit of recoding at intermediate nodes. On the one hand, we construct a suite of cases to show that intermediate recoding is able to bring tremendous benefit (when the size of the network is large), which justifies the necessity of the intermediate recoding. On the other hand, we use numerical evaluation to analysis the benefit in small random networks. The result shows that the benefit in small random networks is modest. An increasing need of intermediate recoding is also suggested as the size of the network grows.

Paper Outline: Section II formulates the flow allocation problem and investigates the properties of the feasible region. Section III considers the feasibility of the problem. Section IV derives a cut-set outer bound on the feasible region, and discusses the tightness of the bound. Section V shows how to find the minimum cost flow using the cut-set bound in some cases. Section VI considers the topology around a vertex, and gives an approach to tighten the cut-set bound. Section VII proposes a route-based flow allocation algorithm in directed acyclic network. Section VIII discusses the benefit of recoding at intermediate nodes. Section IX draws the conclusion.

Notations: Calligraphic letters (such as $\mathcal{X}$ and $\mathcal{Y}$) represent sets and single vertical bars enclosing the sets (such as $|\mathcal{X}|$ and $|\mathcal{Y}|$) are their cardinality. $\mathbb{N}$ is the set of natural numbers. For a set $\mathcal{X}$ and an integer $i\in\mathbb{N}$, define $\mathcal{P}(\mathcal{X},i)=\{\mathcal{X}_1\subseteq\mathcal{X}:|\mathcal{X}_1|\leq i\}$ and $\mathcal{P}(\mathcal{X})=\{\mathcal{X}_1:\mathcal{X}_1\subseteq\mathcal{X}\}$. For an element $x\in\mathcal{X}$ and a set $\mathcal{Y}$, let $x\mathcal{Y}=\{xy:y\in\mathcal{Y}\}$ and $x+\mathcal{Y}=\{x+y:y\in\mathcal{Y}\}$. Boldface letters (such as $\mathbf{x}_{\mathcal{I}}=(x_i:i\in\mathcal{I})$) present indexed lists, whose index sets are denoted by their subscripts. The length of list $\mathbf{x}_{\mathcal{I}}$ is $|\mathbf{x}_{\mathcal{I}}|$. Let $\mathbf{0}_{\mathcal{I}}$ and $\mathbf{1}_{\mathcal{I}}$ be the lists such that

all entries therein are zero and one respectively. An inequality between two lists $\mathbf{x}_{\mathcal{I}}$ and $\mathbf{y}_{\mathcal{I}}$ holds if and only if the inequality $x_i \geq y_i$ hold for every $i \in \mathcal{I}$. For lists $\mathbf{x}_{\mathcal{I}}, \mathbf{y}_{\mathcal{I}} \geq \mathbf{0}_{\mathcal{I}}$, define $\|\mathbf{x}_{\mathcal{I}}\|_1 = \sum_{i\in\mathcal{I}} x_i$ and $\langle \mathbf{x}_{\mathcal{I}}, \mathbf{y}_{\mathcal{I}} \rangle = \sum_{i\in\mathcal{I}} x_i y_i$.

II. Problem Formulation

This paper focuses on link-based attacks in memoryless, noiseless communication networks with a single source node and a single sink node.

In this paper, the network is defined as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any node $v \in \mathcal{V}$, let $\mathrm{In}(v)$ and $\mathrm{Out}(v)$ be the set of incoming and outgoing edges of this vertex respectively. For any directed edge e, let $\mathrm{Tail}(e)$ and $\mathrm{Head}(e)$ be the initial vertex and the terminal vertex of the edge respectively. Let $\{x_{e,1}, x_{e,2}, \cdots\}$ and $\{y_{e,1}, y_{e,2}, \cdots\}$ be the input sequence and the output sequence of the edge e, respectively.

In the process of communication, a message needs to be transmitted from a source node $v_s \in \mathcal{V}$ to a sink node $v_t \in \mathcal{V}$. All nodes in the network cooperate to transmit the message and combat the adversary, and the cooperation scheme is revealed to all parties, including all nodes in the network and the adversary. In addition, the source node v_s and the adversary know the message to be transmitted beforehand.

Although the adversary knows the message to be transmitted and the defending scheme in advance, it can only dominate a portion of edges. Let $\mathcal{A}$ ($\mathcal{A} \subsetneqq \mathcal{P}(\mathcal{E})$) be the set of possible edge sets that the adversary may control. The attacked edge set $\mathcal{E}_a \in \mathcal{A}$ is fixed during the communication. However, before the communication, the legitimate nodes do not know which set in $\mathcal{A}$ is the one that the adversary controls. This error model addresses the worse worst-case scenario (also seen in [22], [23]).

When the flow on edge e is f_e, a symbol in the alphabet $\{1, \cdots, 2^{Nf_e}\}$ can be transmitted in N channel uses, where N is usually assumed to be a very large positive integer [3], [7]. Accordingly, a network error correction flow, henceforth simply called flow and denoted by $\mathbf{f}_{\mathcal{E}} = (f_e : e \in \mathcal{E})$, is a list of edge-flow on $\mathcal{E}$. (Note that, different from the classical network flow, the summation of flows into an intermediate vertex v_i may be unequal to the summation of flows leaving v_i. See Section VI-A for details.) For the sake of simplicity, this paper does not set an upper bound on the edge flows, i.e., each entry in $\mathbf{f}_{\mathcal{E}} = (f_e : e \in \mathcal{E})$ can be arbitrarily large.

Network error correction codes can be constructed on the network error correction flow. Let $R \geq 0$ denote the rate of a code. A $(2^{NR}, N)$ network error correction code consists of

- a message set: $\mathcal{M} = \{1, \cdots, 2^{NR}\}$;
- a set of source encoders, where encoder $e \in \mathrm{Out}(v_s)$ assigns a symbol $x_{e,i}$ to each message $m \in \mathcal{M}$ and v_s's received sequences $(y_e^{i-1} : e \in \mathrm{In}(v_s))$ for $i \in \{1, \cdots, N\}$;
- a set of recoders, where recoder $e \in \mathcal{E}\backslash\mathrm{Out}(v_s)$ assigns a symbols $x_{e,i}$ to each $\mathrm{Tail}(e)$'s received sequences $\left(y_e^{i-1} : e \in \mathrm{In}(\mathrm{Tail}(e))\right)$ for $i \in \{1, \cdots, N\}$;
- a decoder that assigns an estimate $\hat{m}$ to each v_t's received sequences $\left(y_e^N : e \in \mathrm{In}(v_t)\right)$.

A $\left(2^{NR}, N\right)$ network error correction code is on flow $\mathbf{f}_{\mathcal{E}}$ iff $x_e^N, y_e^N \in \left\{1, \cdots, 2^{Nf_e}\right\}$ for every $e \in \mathcal{E}$.

A $\left(2^{NR}, N\right)$ network error correction code is $\mathcal{A}$-error-correcting iff it satisfies the following property: For every edge set $\mathcal{E}_a \in \mathcal{A}$, if $y_e^N = x_e^N$ holds for all $e \in \mathcal{E}\backslash\mathcal{E}_a$, then $\hat{m} = m$.

Fix a rate R and the adversary $\mathcal{A}$. A flow $\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}}$ is a feasible flow if there exists an $\mathcal{A}$-error-correcting $\left(2^{NR}, N\right)$ network error correction code on the flow $\mathbf{f}_{\mathcal{E}}$. Let $\mathcal{F}_{R,\mathcal{A}}$ be the closure of the set of all such feasible flows.

Legitimate users want to use resources as few as possible. Thus, a function from a flow to a non-negative real, called "Cost", is introduced to measure the resources in use. For the sake of simplicity, this paper only considers linear cost functions, whose form is inner product of the unit price and the flow of edges (shown as

$$\mathrm{Cost}(\mathbf{f}_{\mathcal{E}}) = \langle \boldsymbol{\psi}_{\mathcal{E}}, \mathbf{f}_{\mathcal{E}} \rangle = \sum_{e\in\mathcal{E}} \psi_e f_e,$$

where $\boldsymbol{\psi}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}}$ is the unit price of every edge). The minimum cost network error correction flow is the flow that minimizes the cost:

Definition 1 (Minimum Cost Network Error Correction Flow): Use the notations in the preceding contexts. The minimum cost network error correction flow is the optimal solution of the optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathrm{Cost}(\mathbf{f}_{\mathcal{E}}) \\ \text{over} \quad & \mathbf{f}_{\mathcal{E}} \\ \text{s.t.} \quad & \mathbf{f}_{\mathcal{E}} \in \mathcal{F}_{R,\mathcal{A}} \end{aligned} \tag{1}$$

The feasible region of this problem is exactly $\mathcal{F}_{R,\mathcal{A}}$. Some characteristics of this region include:

Proposition 1 (Shape of Feasible Region): Given the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{A}$ is an adversary set and $R > 0$ is the rate. Then

1) (*Homogeneity*) For any $\lambda > 0$, $\mathcal{F}_{\lambda R,\mathcal{A}} = \lambda \mathcal{F}_{R,\mathcal{A}}$;
2) (*Unboundedness*) For any $\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}}$, $\mathcal{F}_{R,\mathcal{A}} + \mathbf{f}_{\mathcal{E}} \subseteq \mathcal{F}_{R,\mathcal{A}}$;
3) (*Convexity*) $\mathcal{F}_{R,\mathcal{A}}$ is convex.

Proof: See Appendix B.[1] □

Since $\mathcal{F}_{R,\mathcal{A}}$ is convex and this paper only considers linear cost function, this optimization problem is a convex optimization problem. However, no exact characterizations of $\mathcal{F}_{R,\mathcal{A}}$ have been derived so far, so the mathematical expression of the feasible region is still unknown. Consequently, this optimization problem can not be solved by using conventional optimization techniques.

III. Feasibility

This section discusses the feasibility of the flow allocation problem. We will show that the feasibility is merely determined by the network topology and the adversary.

[1] Seen from another angle, characterizing $\mathcal{F}$ is a source coding problem, where the rate R is the conventional source rate (usually denoted as $H(U)$ in literatures), and the set $\mathcal{F}$ is the conventional admissible rate region (usually denoted as $\mathscr{R}$ in literatures). So it is not surprising that Proposition 1 is true.

Let $\mathcal{V}_s$ be a set of vertices such that $v_s \in \mathcal{V}_s$ and $v_t \notin \mathcal{V}_s$. A source-sink cut (a.k.a. forward edge set) between $\mathcal{V}_s$ and $\mathcal{V}\backslash\mathcal{V}_s$ is defined as $\mathcal{E}_c^F(\mathcal{V}_s) = \{e \in \mathcal{E} : \text{Tail}(e) \in \mathcal{V}_s, \text{Head}(e) \notin \mathcal{V}_s\}$, while a sink-source cut (a.k.a. backward edge set) between $\mathcal{V}\backslash\mathcal{V}_s$ and $\mathcal{V}_s$ is defined as $\mathcal{E}_c^B(\mathcal{V}_s) = \{e \in \mathcal{E} : \text{Tail}(e) \notin \mathcal{V}_s, \text{Head}(e) \in \mathcal{V}_s\}$. A cut between $\mathcal{V}_s$ and $\mathcal{V}\backslash\mathcal{V}_s$ (bidirectional cut) is defined as $\mathcal{E}_c(\mathcal{V}_s) = \mathcal{E}_c^F(\mathcal{V}_s) \bigcup \mathcal{E}_c^B(\mathcal{V}_s)$.

Theorem 1 (Feasibility): Fix $\mathcal{G}$, $\mathcal{A}$ and $R > 0$. $\mathcal{F}_{R,\mathcal{A}} = \varnothing$ if and only if there exist $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ and a source-sink cut $\mathcal{E}_c^F \subseteq \mathcal{E}$ such that $\mathcal{E}_c^F \subseteq \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}$.

In order to prove Theorem 1, we hereby provide some lemmas in the sequel:

Lemma 1: Fix $\mathcal{G}$, $\mathcal{A}$, and $\mathcal{F}$. If there exist $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ and a source-sink cut $\mathcal{E}_c^F \subseteq \mathcal{E}$ such that $\mathcal{E}_c^F \subseteq \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}$, then $R = 0$.

Let $\mathcal{P}$ be the set of all possible paths[2] (without loop) from the source node v_s to the sink node v_t. For a path $p \in \mathcal{P}$, let $\mathcal{E}_p$ be the set of edges along which path p goes. For an edge set $\mathcal{E}_a \in \mathcal{A}$, let

$$\mathcal{P}_{\mathcal{E}_a} = \left\{p \in \mathcal{P} : \mathcal{E}_p \bigcap \mathcal{E}_a \neq \varnothing\right\}. \tag{2}$$

Lemma 2: Fix $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \subseteq \mathcal{E}$. $\mathcal{P}_{\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}} = \mathcal{P}_{\mathcal{E}_a^{(1)}} \bigcup \mathcal{P}_{\mathcal{E}_a^{(2)}}$.

Lemma 3: Fix $\mathcal{E}_a \subseteq \mathcal{E}$. If $\mathcal{E}_c^F \nsubseteq \mathcal{E}_a$ holds for every source-sink cut $\mathcal{E}_c^F$, there exists a path $p \in \mathcal{P}$ such that $p \notin \mathcal{P}_{\mathcal{E}_a}$.

The proof of Lemma 3 can be found in Appendix C.

Proof of Theorem 1: (1) Due to Lemma 1, when there exist such $\mathcal{E}_a^{(1)}$,$\mathcal{E}_a^{(2)}$, and $\mathcal{E}_c^F$, we have $R=0$. Therefore, $\mathcal{F}_{R,\mathcal{A}}=\varnothing$.

(2) Now we use a proof by contradiction to show that there exist such $\mathcal{E}_a^{(1)}$, $\mathcal{E}_a^{(2)}$, and $\mathcal{E}_c^F$ when $\mathcal{F}_{R,\mathcal{A}} = \varnothing$.

Suppose for arbitrary $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ and every source-sink cut $\mathcal{E}_c^F$, we have $\mathcal{E}_c^F \nsubseteq \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}$. That is, for arbitrary $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ and every source-sink cut $\mathcal{E}_c^F$, there exists an edge e such that $e \in \mathcal{E}_c^F$ and $e \notin \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}$. According to Lemma 2 and Lemma 3, for arbitrary $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$, there exists a path $p \notin \mathcal{P}_{\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}} = \mathcal{P}_{\mathcal{E}_a^{(1)}} \bigcup \mathcal{P}_{\mathcal{E}_a^{(2)}}$, i.e., $p \notin \mathcal{P}_{\mathcal{E}_a^{(1)}}$ and $p \notin \mathcal{P}_{\mathcal{E}_a^{(2)}}$. Let $\bar{p}_{\mathcal{E}_a^{(1)},\mathcal{E}_a^{(2)}}$ denote this path.

Given $\mathcal{E}_a \in \mathcal{A}$, let $\bar{\mathcal{P}}_{\mathcal{E}_a} = \left\{\bar{p}_{\mathcal{E}_a,\mathcal{E}_a^{(2)}} : \mathcal{E}_a^{(2)} \in \mathcal{A}\right\}$. For an arbitrary $\mathcal{E}_a^{(2)} \in \mathcal{A}$, due to $\bar{p}_{\mathcal{E}_a,\mathcal{E}_a^{(2)}} \in \bar{\mathcal{P}}_{\mathcal{E}_a}$ and $\bar{p}_{\mathcal{E}_a,\mathcal{E}_a^{(2)}} \notin \mathcal{P}_{\mathcal{E}_a^{(2)}}$, we have $\bar{\mathcal{P}}_{\mathcal{E}_a} \nsubseteq \mathcal{P}_{\mathcal{E}_a^{(2)}}$.

Consider the following network error correction code: For each path $p \in \mathcal{P}$, send the message directly. That is, the message is repeatedly sent $|\mathcal{P}|$ times. According to the above analysis, on the one hand, since at most one set within $\mathcal{A}$ is controlled by the adversary, there exists a set of paths, say $\bar{\mathcal{P}}_{\mathcal{E}_a}$, that are not controlled by the adversary; on the other hand, no adversary can totally change the signals in $\bar{\mathcal{P}}_{\mathcal{E}_a}$. Therefore, the sink node can always find a set of paths $\bar{\mathcal{P}}_{\mathcal{E}_a}$ ($\mathcal{E}_a \subseteq \mathcal{A}$) where the received signals are both identical and correct. In this way, we have constructed a code that supports rate R and combats adversary $\mathcal{A}$ simultaneously. It contradicts $\mathcal{F}_{R,\mathcal{A}} = \varnothing$ and completes the proof. □

[2]Here, the stand-alone symbol $\mathcal{P}$ denotes the set of paths from v_s to v_t, which differs from the uniary/binary function (in the form of $\mathcal{P}(\mathcal{X})$ or $\mathcal{P}(\mathcal{X}, i)$) defined in Section I.

IV. Cut-Set Bound on $\mathcal{F}_{R,\mathcal{A}}$

A. Cut-Set Outer Bound on Feasible Region

This section considers an outer bound on the feasible region $\mathcal{F}_{R,\mathcal{A}}$, which uses a cut $\mathcal{E}_c$ with forward edge set (source-sink cut) $\mathcal{E}_c^F \subseteq \mathcal{E}_c$ and backward edge set (sink-source cut) $\mathcal{E}_c^B \subseteq \mathcal{E}_c$ to restrict the feasible region $\mathcal{F}_{R,\mathcal{A}}$.

Theorem 2 (Cut-set Outer Bound): Given $R > 0$ and $\mathcal{A} \subsetneqq \mathcal{P}(\mathcal{E})$ in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{E}_c \subseteq \mathcal{E}$ be a cut in the network, and its forward edge set is $\mathcal{E}_c^F$ and its backward edge set is $\mathcal{E}_c^B$. Define

$$\mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right) = \bigcap_{\mathcal{E}_a \in \mathcal{A}} \left\{\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}} : \|\mathbf{f}_{\mathcal{E}_c^F \backslash \mathcal{E}_a}\|_1 \geq R\right\},$$

$$\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right) = \bigcap_{\substack{\mathcal{E}_a^{(1)} \in \mathcal{A}: \mathcal{E}_c^B \subseteq \mathcal{E}_a^{(1)} \\ \mathcal{E}_a^{(2)} \in \mathcal{A}: \mathcal{E}_c^B \subseteq \mathcal{E}_a^{(2)}}} \left\{\mathbf{f}_{\mathcal{E}} \geqq \mathbf{0}_{\mathcal{E}} : \left\|\mathbf{f}_{\mathcal{E}_c^F \backslash \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}\right\|_1 \geq R\right\},$$

and

$$\mathcal{B}_{R,\mathcal{A}}(\mathcal{E}_c) = \begin{cases} \varnothing, & \text{if } \exists \mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A} \text{ such that } \mathcal{E}_c^F \subseteq \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)} \\ \mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right) \bigcap \mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right), & \text{otherwise,} \end{cases}$$

then

$$\mathcal{F}_{R,\mathcal{A}} \subseteq \mathcal{B}_{R,\mathcal{A}}(\mathcal{E}_c).$$

The proof of Theorem 2 is provided in Appendix D.

Interpretation of the Cut-Set Outer Bound: This cut-set bound tells us, when there exist $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ such that $\mathcal{E}_c^F \subseteq \mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}$, $\mathcal{F}_{R,\mathcal{A}}$ is an empty set; otherwise, $\mathcal{F}_{R,\mathcal{A}}$ is bounded by both $\mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right)$ and $\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right)$. The set $\mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right)$ can be understood as follows: Consider a feasible flow $\mathbf{f}_{\mathcal{E}} \in \mathcal{F}_{R,\mathcal{A}}$ and an edge set $\mathcal{E}_a \in \mathcal{A}$. When the adversary controls $\mathcal{E}_a$, the edge flow on $\mathcal{E}_a$ does not help the communication. In this sense, the remaining flow, i.e. $\mathbf{f}_{\mathcal{E}\backslash\mathcal{E}_a}$, should be able to support the communication. In this sense, the rate R can not exceed the summation of flow $\mathbf{f}_{\mathcal{E}\backslash\mathcal{E}_a}$ on the forward cut $\mathcal{E}_c^F$, which results in

$$\left\|\mathbf{f}_{\mathcal{E}_c^F \backslash \mathcal{E}_a}\right\|_1 \geq R.$$

Similarly, the set $\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right)$ can be understood as follows: Consider a cut $\mathcal{E}_c(\mathcal{V}_s) = \mathcal{E}_c^F \bigcup \mathcal{E}_c^B$. If there are no $\mathcal{E}_a \in \mathcal{A}$ such that $\mathcal{E}_c^B \subseteq \mathcal{E}_a$, $\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right) = \{\mathbf{f}_{\mathcal{E}} : \mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}}\}$ is a trivial bound. If there exists a unique $\mathcal{E}_a \in \mathcal{A}$ such that $\mathcal{E}_c^B \subseteq \mathcal{E}_a$, $\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right) = \left\{\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}} : \|\mathbf{f}_{\mathcal{E}_c^F \backslash \mathcal{E}_a}\|_1 \geq R\right\}$ is a subset of $\mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right)$, which is trivial too. If there are multiple $\mathcal{E}_a \in \mathcal{A}$ such that $\mathcal{E}_c^B \subseteq \mathcal{E}_a$, let $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)}$ denote two different such sets. Either of the two sets is able to completely forbid any meaningful information from $\mathcal{V}\backslash\mathcal{V}_s$ to $\mathcal{V}_s$. Therefore, the flow on $\mathcal{E}_c^F$ shall use a forward error correction code to ensure the correctness of the communication, which requires

$$\left\|\mathbf{f}_{\mathcal{E}_c^F \backslash \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}\right\|_1 \geq R.$$

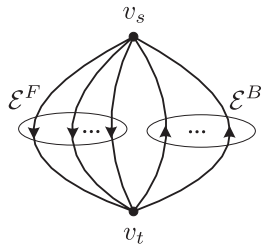


Fig. 2. A two node network.

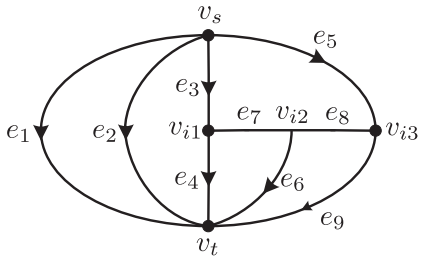


Fig. 3. An example where the cut-set outer bound is not tight.

Now we consider a simplified case that the adversary corrupts the symbols on arbitrary z edges ($z \in \mathbb{N}$). Let $\mathcal{A} = \mathcal{P}(\mathcal{E}, z)$, and the cut-set outer bound reduces to [3, Theorem 2]

$$\mathcal{B}_{R,\mathcal{P}(\mathcal{E},z)}(\mathcal{E}_c) = \begin{cases} \varnothing, & \left|\mathcal{E}_c^F\right| \le 2z \\ \left\{\mathbf{f}_{\mathcal{E}} \ge \mathbf{0}_{\mathcal{E}} : \min\limits_{\mathcal{E}_a \in \mathcal{P}(\mathcal{E}_c^F, n_z)} \left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \mathcal{E}_a}\right\|_1 \ge R\right\}, & \text{otherwise,} \end{cases} \tag{3}$$

where

$$n_z = \max\left\{z, 2\left(z - \left|\mathcal{E}_c^B\right|\right)^+\right\}. \tag{4}$$

B. Tightness of the Cut-Set Outer Bound

Let $\mathcal{B}_{R,\mathcal{A}}$ denote the outer bound obtained by applying Theorem 2 in all cuts, i.e.,

$$\mathcal{B}_{R,\mathcal{A}} = \bigcap_{\mathcal{V}_s \subseteq \mathcal{V}: v_s \in \mathcal{V}_s, v_t \notin \mathcal{V}_s} \mathcal{B}_{R,\mathcal{A}}\left(\mathcal{E}_c(\mathcal{V}_s)\right).$$

This subsection discusses whether $\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}}$.

According to Theorem 1 and Theorem 2, when $\mathcal{F}_{R,\mathcal{A}} = \varnothing$, $\mathcal{B}_{R,\mathcal{A}} = \varnothing$. In this trivial case, $\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}}$. In fact, $\mathcal{B}_{R,\mathcal{A}} = \mathcal{F}_{R,\mathcal{A}}$ may also hold for other cases. Here, we provide an example where $\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}} \neq \varnothing$.

Example 2: Consider the two-node network [3] in Fig. 2. R is a positive real number and z is a positive integer. There are $n^F > 2z$ edges from v_s to v_t (collectively denoted as $\mathcal{E}^F$), and $n^B \ge 0$ edges from v_t to v_s (collectively denoted as $\mathcal{E}^B$). Then

$$\mathcal{F}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)} = \mathcal{B}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)}$$

Proof: Here we only consider cut $\mathcal{E} = \mathcal{E}^F \bigcup \mathcal{E}^B$ and prove

$$\mathcal{B}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)} \subseteq \mathcal{F}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)}.$$

For any $\mathbf{f}^*_{\mathcal{E}^F \bigcup \mathcal{E}^B} \in \mathcal{B}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)}$,

$$\left\|\mathbf{f}^*_{\mathcal{E}^F \setminus \mathcal{E}_a}\right\|_1 \ge R, \quad \mathcal{E}_a \in \mathcal{P}\left(\mathcal{E}^F, n_z\right)$$

where n_z is defined in equation (4). Now we prove

$$\mathbf{f}^*_{\mathcal{E}^F \bigcup \mathcal{E}^B} \in \mathcal{F}_{R,\mathcal{P}(\mathcal{E}^F \bigcup \mathcal{E}^B, z)}. \tag{5}$$

Consider the flow

$$\mathbf{f}^{**}_{\mathcal{E}^F \bigcup \mathcal{E}^B} = \mathbf{f}^*_{\mathcal{E}^F \bigcup \mathcal{E}^B} + \varepsilon \mathbf{1}_{\mathcal{E}^F \bigcup \mathcal{E}^B},$$

where ε is an arbitrary positive real number. We can verify that for any $\mathcal{E}_a \in \mathcal{P}\left(\mathcal{E}^F, n_z\right)$,

$$\left\|\mathbf{f}^{**}_{\mathcal{E}^F \setminus \mathcal{E}_a}\right\|_1 > R.$$

According to [3, Theorem 2], a $\left(2^{NR}, N\right)$ network error correction code can be constructed on flow $\mathbf{f}^{**}_{\mathcal{E}^F \bigcup \mathcal{E}^B}$. That is, flow $\mathbf{f}^*_{\mathcal{E}^F \bigcup \mathcal{E}^B} + \varepsilon \mathbf{1}_{\mathcal{E}^F \bigcup \mathcal{E}^B}$ is a feasible flow. Let $\varepsilon \to 0$, which leads to (5). The proof is completed. □

This example shows that the cut-set outer bound in Theorem 2 is the tightest one among all bounds without considering the topology outside the cut $\mathcal{E}_c$.

Unfortunately, although Theorem 2 provides the tightest cut-based outer bound, this bound is not tight in general. In order to show this, we provide an example where the cut-set outer bound is not tight.

Example 3: In Fig. 3, we want to support the rate $R > 0$ and combat the adversary $\mathcal{A} = \mathcal{P}(\mathcal{E}, 1)$. Consider the flow $\mathbf{f}_{\mathcal{E}} = (f_e : e \in \mathcal{E})$:

$$f_e = \begin{cases} \frac{1}{2}R, & e \in \{e_1, e_2, \cdots, e_6\} \\ \varepsilon, & e \in \{e_7, e_8, e_9\} \end{cases} \tag{6}$$

where ε is a very small positive real such that $\varepsilon \ll \frac{1}{4}R$. It can be verified that $\mathbf{f}_{\mathcal{E}} \in \mathcal{B}_{R,\mathcal{A}}$.

For intermediate node v_{i2}, the only incoming flow is ε on edge e_7. The signals in e_7 completely determine the signals in e_6, so the flow on edge e_6 carries at most ε information. Considering this in cut $\{e_1, e_2, e_4, e_6, e_9\}$, the supported rate could not be greater than $\frac{1}{2}R + 2\varepsilon$. Consider that $\varepsilon \ll \frac{1}{4}R$, rate R is impossible to achieve. That is,

$$\mathbf{f}_{\mathcal{E}} \notin \mathcal{F}_{R,\mathcal{A}}.$$

Therefore, the cut-set outer bound is not tight in this example.

C. Generalize to Multicast

Theorem 2 can easily extended into the multicast scenario: When there are multiple sink nodes (collectively denoted as a set $\mathcal{D}$ such that $\mathcal{D} \subseteq \mathcal{V} \setminus \{v_s\}$), we can obtain a outer bound on

the feasible region by applying the aforementioned unicast cut-set bound on each sink node, i.e.,

$$\mathcal{B}_{R,\mathcal{A}}^{(v_t)} = \bigcap_{\mathcal{V}_s \subseteq \mathcal{V}: v_s \in \mathcal{V}_s, v_t \notin \mathcal{V}_s} \mathcal{B}_{R,\mathcal{A}}\left(\mathcal{E}_c(\mathcal{V}_s)\right), \quad v_t \in \mathcal{D}.$$

Obviously, the feasible region is outer bounded by

$$\mathcal{F}_{R,\mathcal{A}} \subseteq \bigcap_{v_t \in \mathcal{D}} \mathcal{B}_{R,\mathcal{A}}^{(v_t)}.$$

V. Find the Minimum Cost Network Error Correction Flow: A Cut-Based Approach

This section presents an approach to obtain minimum cost network error correction flows in some cases. Firstly, we use the cut-set outer bound in the previous section to relax the original flow allocation problem. Using the relaxed problem, we can find a lower bound on the minimum cost, and an optimal solution of the optimization. Next, we check whether the optimal solution of the relaxed problem is a feasible flow. If we can construct a network error correction scheme on that flow, the flow is the minimum cost network error correction flow.

A. Cut-Based Lower Bound on Minimum Cost

Using the cut-set outer bound in Theorem 2, the original optimization problem 2 can be relaxed to the following problem:

$$\begin{aligned} \text{minimize} \quad & \text{Cost}(\mathbf{f}_{\mathcal{E}}) \\ \text{over} \quad & \mathbf{f}_{\mathcal{E}} \\ \text{s.t.} \quad & \mathbf{f}_{\mathcal{E}} \in \mathcal{B}_{R,\mathcal{A}}. \end{aligned} \tag{7}$$

Obviously, the optimal value of the slacked problem (7) is a lower bound on the minimum cost.

Note that, for each cut $\mathcal{E}_c$, both $\mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right)$ and $\mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right)$ are intersections of half spaces, which implies that $\mathcal{B}_{R,\mathcal{A}}(\mathcal{E}_c)$ is actually an unbounded polyhedron. In this sense, with linear cost function, the slacked problem (7) is actually a linear programming. Through this method, we successfully find a computable lower bound on the minimum cost.

Additionally, since Example 2 shows that the cut-set bound in Theorem 2 is the tightest outer bound among all outer bounds without considering the topology outside the cut $\mathcal{E}_c$, the relaxed problem (7) provides the tightest cut-based lower bound.

B. Examples of Finding Minimum Cost Network Error Correction Flow

The cut-based lower bound can help us to find the minimum cost of network error correction flow in many cases, even when the cut-set outer bound is not exactly the feasible region. Hereby, we present two instances to show how to find the minimum cost network error correction flow using the cut-based lower bound.

Example 4: For the two networks in Fig. 4 (a) and (b), $\boldsymbol{\psi}_{\mathcal{E}} = \mathbf{1}_{\mathcal{E}}$, $\mathcal{A} = \mathcal{P}(\mathcal{E}, 1)$. The message rate $R > 0$ is given, too. For either graph, its edges can be classified into three layers: The upper layer of edges $\mathcal{E}_u = \text{Out}(v_s)$ contains the edges that are connected with v_s; the bottom layer of edges $\mathcal{E}_b = \text{In}(v_t)$ contains the edges that are connected with v_t; and the intermediate layer of edges $\mathcal{E}_m = \mathcal{E} \backslash \mathcal{E}_u \backslash \mathcal{E}_b$. The set $\mathcal{V}_u$ consists of the nodes connected with $\mathcal{E}_u$, i.e. $\mathcal{V}_u = \bigcup_{e \in \mathcal{E}_u} \text{Tail}(e)$. The set $\mathcal{V}_b$ consists of the nodes connected with $\mathcal{E}_b$, i.e. $\mathcal{V}_b = \bigcup_{e \in \mathcal{E}_b} \text{Head}(e)$. Now we try to find the minimum cost network error correction flow in these two instances.

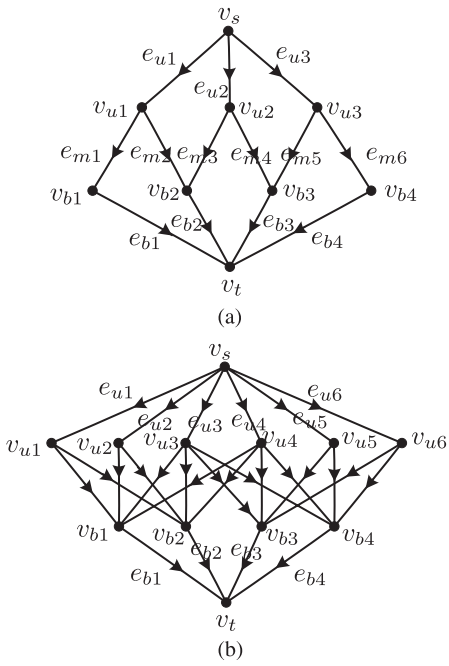


Fig. 4. Two graphs with identical unit price. (a) A direct graph with $3 + 6 + 4 = 13$ edges. (b) A direct graph with $6 + 16 + 4 = 26$ edges.

(a) Firstly, we show that the cost of any feasible flow is not less than $8R$. Since any $z = 1$ edge can be malicious and no backwards edges exist in this cut, equation (4) reduces to $n_z = 2$. For any feasible flow $\mathbf{f}_{\mathcal{E}}$, apply the cut-set bound (3) to $\mathcal{E}_u$, and we will get

$$f_{e_{uj}} \geq R, \quad j = 1, 2, 3,$$

resulting in

$$\|\mathbf{f}_{\mathcal{E}_u}\|_1 \geq 3R.$$

Apply the cut-set bound to

$$\mathcal{E}_{cj} = \left\{e : \text{Tail}(e) \in \{v_s, v_{uj}\}, \text{Head}(e) \notin \{v_s, v_{uj}\}\right\} \quad j = 1, 2, 3,$$

and we will get

$$\|\mathbf{f}_{\text{Out}(v_{uj})}\| \geq R, \quad j = 1, 2, 3,$$

which results in

$$\|\mathbf{f}_{\mathcal{E}_i}\|_1 \geq 3R.$$

Apply the cut-set bound to $\mathcal{E}_b$ and we will get

$$\|\mathbf{f}_{\mathcal{E}_b}\|_1 \geq 2R.$$

The cost of any feasible flow is lower bounded by

$$\text{Cost}(\mathbf{f}_{\mathcal{E}}) = \|\mathbf{f}_{\mathcal{E}_u}\|_1 + \|\mathbf{f}_{\mathcal{E}_m}\|_1 + \|\mathbf{f}_{\mathcal{E}_b}\|_1 \geq 3R + 3R + 2R = 8R.$$

Note that [3, Lemma 7] found a code on the flow

$$\left(\mathbf{f}^{\text{opt}}_{\mathcal{E}_u}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_m}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_b}\right) = \left(R\mathbf{1}_{\mathcal{E}_u}, \frac{1}{2}R\mathbf{1}_{\mathcal{E}_m}, \frac{1}{2}R\mathbf{1}_{\mathcal{E}_b}\right),$$

whose cost is exactly $8R$, and this code can combat any one erroneous link and support rate R. Therefore, the flow is a feasible flow, and consequently, the minimum cost flow.

(b) Similar to the proof in (a), we firstly show that the cost of any feasible flow is not less than $\frac{11}{2}R$.

Now we apply the cut-set bound to $\mathcal{E}_u$. Since any $z = 1$ edge can be malicious and no backwards edges exist in this cut, equation (4) reduces to $n_z = 2$. Due to equation (3), we can obtain the following inequalities

$$f_{e_{u1}} + f_{e_{u2}} + f_{e_{u3}} + f_{e_{u4}} \geq R,$$

$$f_{e_{u1}} + f_{e_{u2}} + f_{e_{u5}} + f_{e_{u6}} \geq R,$$

$$f_{e_{u3}} + f_{e_{u4}} + f_{e_{u5}} + f_{e_{u6}} \geq R.$$

Add the three inequalities together, and it will result in

$$\|\mathbf{f}_{\mathcal{E}_u}\|_1 \geq \frac{3}{2}R.$$

Apply the cut-set bound to

$$\mathcal{E}_{c1} = \{e : \text{Tail}(e) \notin \{v_{b1}, v_{b2}, v_t\}, \text{Head}(e) \in \{v_{b1}, v_{b2}, v_t\}\},$$

then

$$\left\|\mathbf{f}_{\text{In}(v_{b1})}\right\|_1 + \left\|\mathbf{f}_{\text{In}(v_{b2})}\right\|_1 \geq R.$$

Similarly, we can prove

$$\left\|\mathbf{f}_{\text{In}(v_{b3})}\right\|_1 + \left\|\mathbf{f}_{\text{In}(v_{b4})}\right\|_1 \geq R.$$

Thus,

$$\begin{aligned}\left\|\mathbf{f}_{\mathcal{E}_m}\right\|_1 &= \left\|\mathbf{f}_{\text{In}(v_{b1})}\right\|_1 + \left\|\mathbf{f}_{\text{In}(v_{b2})}\right\|_1 + \left\|\mathbf{f}_{\text{In}(v_{b3})}\right\|_1 + \left\|\mathbf{f}_{\text{In}(v_{b4})}\right\|_1 \\ &\geq 2R.\end{aligned}$$

Consider the cut $\mathcal{E}_b$,

$$f_{b_1} + f_{b_2} \geq R,$$

$$f_{b_3} + f_{b_4} \geq R,$$

then

$$\|\mathbf{f}_{\mathcal{E}_b}\|_1 \geq 2R.$$

Thus, the expense of a feasible flow is lower bounded by

$$\begin{aligned}\text{Cost}(\mathbf{f}_{\mathcal{E}}) &= \|\mathbf{f}_{\mathcal{E}_u}\|_1 + \|\mathbf{f}_{\mathcal{E}_m}\|_1 + \|\mathbf{f}_{\mathcal{E}_b}\|_1 \\ &\geq \frac{3}{2}R + 2R + 2R \\ &= \frac{11}{2}R.\end{aligned}$$

Next, we show that the flow

$$\left(\mathbf{f}^{\text{opt}}_{\mathcal{E}_u}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_m}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_b}\right) = \left(\frac{1}{4}R\mathbf{1}_{\mathcal{E}_u}, \frac{1}{8}R\mathbf{1}_{\mathcal{E}_m}, \frac{1}{2}R\mathbf{1}_{\mathcal{E}_b}\right),$$

whose cost is exactly $\frac{11}{2}R$, is a feasible flow. Here we use letters (such as a, b, ...) to represent $R/8$-flow. The $R/8$-flows on the same edge are separated by commas, and the $R/8$-flows on different edges are separated by semicolons. Let the edges $\mathcal{E}_u$ transmit a (12, 8) maximum-distance-separable (MDS) code [24] akin to (a,b; c,d; e,f; g,h; i,j; k,l). The edges $\text{Out}(v_{u1}), \text{Out}(v_{u2}), \cdots, \text{Out}(v_{u6})$ transmit streams (a; b), (c; d), (e; f; e + f; e + 2f), (g; h; g + h; g + 2h), (i; j), (k; l) respectively, where additions are executed in the finite field $\mathbb{F}_q$ (q is a large positive integer). The nodes in $\mathcal{V}_b$ forward all their received streams to v_t, in the form of (a, c, e, g; b, d, f, g; e + f, g + h, i, k; e + 2f, g + 2h, k, l). Node v_t decodes in two steps: First, it decodes the streams related to e, f, g and h by the method to decode a (4, 2) MDS code. Second, v_t decodes output in first step (a,b,c,d,e,f,g,h) by the method to decode a (12, 8) MDS code. Now we show that the code can correct arbitrary one error. If one of edges in $\mathcal{E}_u$ is tampered with, the number of incorrect streams after the first decode step are not greater than two and can be corrected in the second decode step. If one of edges in $\mathcal{E}_i$ errors, the number of error stream received by v_t is at most one and can be corrected by either decode steps. If one of the edges in $\mathcal{E}_b$ is tampered with, the streams related to e, f, g and h can be corrected by the (4, 2) MDS code in the first decode step, while the remaining two streams can be corrected by the (12, 8) MDS code. Thus, $\left(\mathbf{f}^{\text{opt}}_{\mathcal{E}_u}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_m}, \mathbf{f}^{\text{opt}}_{\mathcal{E}_b}\right) = \left(\frac{1}{4}R\mathbf{1}_{\mathcal{E}_u}, \frac{1}{8}R\mathbf{1}_{\mathcal{E}_m}, \frac{1}{2}R\mathbf{1}_{\mathcal{E}_b}\right)$ is a feasible flow, and consequently, the minimum cost flow.

Comparison Between Fig. 4 (a) and (b): The minimum cost in Fig. 4 (a) and (b) are $8R$ and $\frac{11}{2}R$, respectively. Because the network in Fig. 4(b) has better connectivity than the network in Fig. 4(a), the minimum cost in Fig. 4(b) is lower than that in Fig. 4(a). This result implies that the network with better connectivity may obtain lower minimum cost.

C. Relationship Between Tightness of Cut-Set Bound on Feasible Region and Cut-Based Bound on Minimum Cost

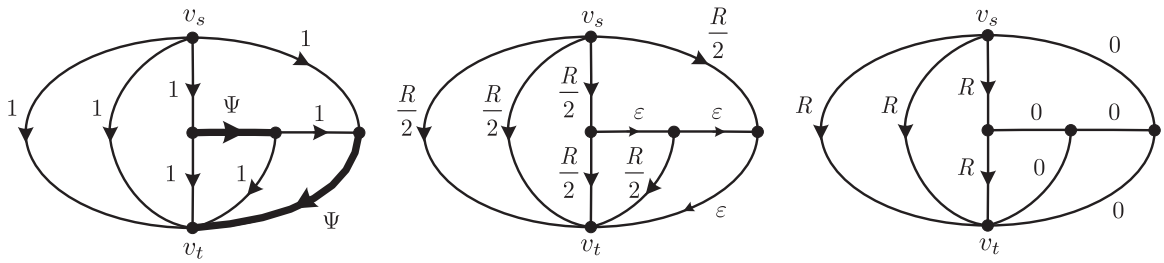
Obviously, the tightness of the cut-set outer bound on the feasible region has direct impact on the tightness of the cut-based lower bound on the minimum cost in (7) (with respect to the original optimization problem (1)).

Proposition 2: The relaxed problem (7) is tight for all cost functions if and only if

$$\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}}.$$

Proof: (1) If the relaxed problem (7) is tight for all cost functions, the relaxed problem is tight for all linear cost functions. This in fact tells us $\mathcal{F}_{R,\mathcal{A}}$ and $\mathcal{B}_{R,\mathcal{A}}$ share identical supporting hyperplanes. Recalling that Proposition 1 tells us $\mathcal{F}_{R,\mathcal{A}}$ is unbounded and convex, it is obvious that $\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}}$.

(2) If $\mathcal{F}_{R,\mathcal{A}} = \mathcal{B}_{R,\mathcal{A}}$, the relaxed problem (7) is equivalent to the original problem (1). Therefore, the relaxed problem is tight whatever the cost function is. □



(a) unit price of each edge, $\Psi \ll 1$ (b) an infeasible flow (cost $\approx 3R$) (c) a feasible flow (cost $4R$)

Fig. 5. Example 5. All vertices and edges are the same as Fig. 3. (a) Shows the unit price of each edge. (b) Shows the solution of the relaxed problem (7), which is infeasible. (c) Shows the solution of the problem (8), which is a minimum cost network error correction flow.

VI. More than Cut-Set Bound

The cut-based approach in the previous section does not always find the minimum cost network error correction flow since the cut-set outer bound is not tight in general and the tightness of cut-set outer bound directly relates to the tightness of the cut-based relaxed problem (7). Considering this point, this section tries to improve the lower bound by adding more constraints on the relaxed problem.

A. Property of Minimum Cost Network Error Correction Flow

This subsection considers the topology around a vertex, and try to find a property of minimum cost network correction flow.

For a vertex $v \in \mathcal{V}\backslash\{v_s\}$, the input signals of this vertex completely determine the outputs of this vertex, i.e., $\mathrm{In}(v)$ *informationally dominates* $\mathrm{Out}(v)$ (The concept of informational dominance was formally proposed in [25]). In this sense, an edge $e \in \mathcal{E}\backslash\mathrm{Out}(v_s)$ need not have more than $\mathrm{Tail}(e)$'s incoming flows. This observation can be formally stated as the following proposition:

Proposition 3: Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\boldsymbol{\psi}_{\mathcal{E}}$, $\mathcal{A}$, and R, $\mathbf{f}_{\mathcal{E}}^{\mathrm{opt}} = \left(f_e^{\mathrm{opt}} : e \in \mathcal{E}\right)$ is a minimum cost network error correction flow. For any $e \in \mathcal{E}\backslash\mathrm{Out}(v_s)$ such that $\psi_e > 0$,

$$f_e^{\mathrm{opt}} \leq \left\| \mathbf{f}^{\mathrm{opt}}_{\mathrm{In}(\mathrm{Tail}(e))} \right\|_1.$$

This result reveals the relationship between incoming edge flows and outgoing edge flows of a vertex. In contrast, in the case of classical network flows, which are unable to resist errors, for a particular vertex (except the source node or the sink node), the summation of all incoming edge flows always equals the summation of all outgoing edge flows. This property is well known as the law of flow conservation. Unfortunately, the property no longer holds for network error correction flows in general. In this sense, Proposition 3 can be regarded as an extended version of the flow conservation.

B. Tighter Bound

According to the aforementioned proposition, a relaxed problem is defined as follows to provide a tighter lower bound when we try to find the minimum cost network error correction flow. This new lower bound is sometimes tighter than the cut-based lower bound in (7).

$$\begin{aligned} \text{minimize} \quad & \mathrm{Cost}(\mathbf{f}_{\mathcal{E}}) \\ \text{over} \quad & \mathbf{f}_{\mathcal{E}} \\ \text{s.t.} \quad & \mathbf{f}_{\mathcal{E}} \in \mathcal{B}_{R,\mathcal{A}} \\ & f_e \leq \left\| \mathbf{f}_{\mathrm{In}(\mathrm{Tail}(e))} \right\|_1, \quad e \in \mathcal{E}\backslash\mathrm{Out}(v_s). \end{aligned} \tag{8}$$

Example 5: Let us reconsider the network in Example 3. In this example, the unit price of edge flow in each edge is

$$\psi_e = \begin{cases} 1, & e \notin \{e_7, e_9\} \\ \Psi, & e \in \{e_7, e_9\} \end{cases}$$

where Ψ is a very large positive real number. In this condition, the optimal solution of problem (7) is exactly equation (6), which proved to be infeasible in Example 3. The optimal solution of problem (8) is

$$f_e^{\mathrm{rel}} = \begin{cases} R, & e \in \{e_1, e_2, e_3, e_4\} \\ 0, & e \notin \{e_1, e_2, e_3, e_4\}. \end{cases}$$

This flow is a feasible flow, since we can construct a (3,1) MDS code therein. Therefore, this flow is the minimum cost network error correction flow (see Fig. 5.)

VII. Optimal Allocation Without Intermediate Recoding in DAG

In this section, we propose an algorithm to allocate network error correction flow on directed acyclic graphs (DAG) when $\mathcal{A} = \mathcal{P}(\mathcal{E}, z)$.

In a directed acyclic graph $\mathcal{G}$, the maximum number of disjoint paths (denoted as $|\mathcal{P}|$) is equal to the minimum cardinality of all cuts in the graph. There are no backward edges. When intermediate recoding is forbidden (as the case in real life that routers are unable to operate data), network error correction flow should be allocated in disjoint paths.

A. Preliminary

In our previous paper [21], we proved that the best way to allocate flow in disjoint paths is to distribute edge-flow equally when the problem is feasible.

Theorem 3 (Theorem 1 of [21]): In a network that consists of $|\mathcal{P}|$ disjoint paths, the Cost function is $\text{Cost}(\mathbf{f}_{\mathcal{P}}) = \sum_{i=1}^{|\mathcal{P}|} \psi_i f_i$, where $\psi_1 \leq \psi_2 \leq \cdots \leq \psi_{|\mathcal{P}|}$. n_z has been determined by equation (4). Let

$$n_r = \arg\max_{1 \leq i \leq |\mathcal{P}|} \frac{\max\{i - n_z, 0\}}{\sum_{j=1}^{i} \psi_j} - n_z.$$

When $n_r > 0$, the problem is feasible, and the flow

$$\mathbf{f}_{\mathcal{P}}^{\text{opt}} = \left\{ \frac{R}{n_r} \mathbf{1}_{n_r+n_z}, \mathbf{0}_{|\mathcal{P}|-(n_r+n_z)} \right\}$$

is the minimum cost network error correction flow; otherwise, the problem is infeasible.

At the same time, [21, Lemma 1] provides a criterion to determine the number of paths to allocate edge-flows.

Lemma 4 (Lemma 2 of [21]): In the context of Theorem 3, let

$$\mathcal{I} = \{1 \leq i \leq |\mathcal{P}| : \phi_{i-1} \leq (i - 1 - n_z)\psi_i\}$$

where

$$\phi_i = \begin{cases} 0, & i = 0 \\ \sum_{j=1}^{i} \psi_j, & 1 \leq i \leq |\mathcal{P}|. \end{cases}$$

Then

$$n_r + n_z = \begin{cases} \min_{i \in \mathcal{I}} i, & \mathcal{I} \neq \varnothing \\ |\mathcal{P}|, & \mathcal{I} = \varnothing. \end{cases}$$

Lemma 4 provides a method to determine the value of n_r efficiently. In [21], we proposed an algorithm (see Fig. 6) to find the value of n_r using a loop. In the i-th iteration of the loop ($1 \leq i \leq |\mathcal{P}|$), we compare the costs between allocating the flow in i paths and in $(i-1)$ paths by evaluating $\phi_{i-1} \leq (i - 1 - n_z)\psi_i$, and update the value of ϕ_i by assigning $\phi_i \leftarrow \phi_{i-1} + \psi_i$. Once we find an i such that $\phi_{i-1} \leq (i - 1 - n_z)\psi_i$, let $n_r \leftarrow i$. If for all i, $\phi_{i-1} > (i - 1 - n_z)\psi_i$, then $n_r + n_z = |\mathcal{P}|$.

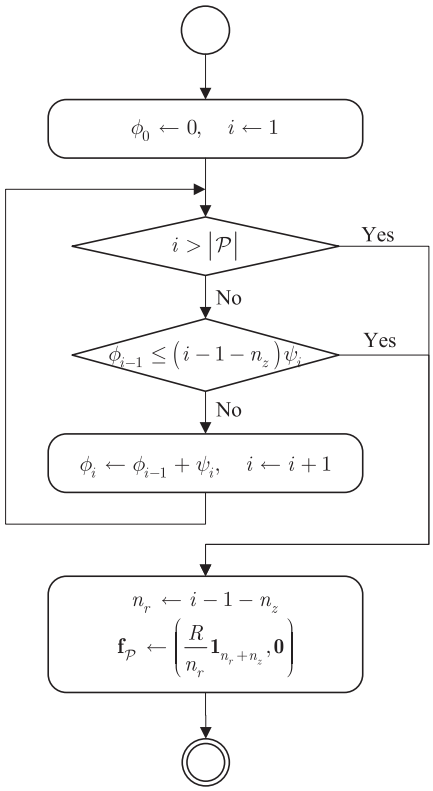


Fig. 6. Flow chart of finding min-cost network error correction flow on disjoint paths.

B. Algorithm

In this subsection, we propose an algorithm to find disjoint minimum cost network error correction flow in DAG when $\mathcal{A} = \mathcal{P}(\mathcal{E}, z)$. (see Algorithm 1).

Algorithm 1 Allocate the minimum cost flow without intermediate recoding

Input: $\boldsymbol{\psi}_{\mathcal{E}} = (\psi_e, e \in \mathcal{E})$, n_z, R.
Output: $\mathbf{f}_{\mathcal{E}}^{\text{fwd}} = (f_e^{\text{fwd}} : e \in \mathcal{E})$, c^{fwd}.
1: **Initialize**: Calculate n_z using equation (4). $n_r \leftarrow (-n_z)$. $\phi \leftarrow 0$.
2: **loop**
3: Try to find one of the shortest paths from v_s to v_t (using the shortest path algorithm which can trickle negative edges).
4: **if** no path can be found **then**
5: Break the loop;
6: **else**
7: $\psi \leftarrow$ the summation of all unit prices of edges on the path just found.
8: **end if**
9: **if** $\phi \leq n_r\psi$ **then**
10: Break the loop;
11: **else**
12: $\phi \leftarrow \phi + \psi$, $n_r \leftarrow n_r + 1$.
13: Inverse the edges on the path just found with negative unit prices. (Note that if there are multiple identical edges between two vertices, only one edge needs to be revised.)
14: **end if**
15: **end loop**
16: **if** $n_r > 0$ **then**
17: $\mathbf{f}_{\mathcal{E}}^{\text{fwd}}$: Allocate $\frac{R}{n_r}$ on all the $n_z + n_r$ inverse paths from v_t to v_s.
Other edges will not be used.
18: c^{fwd}: $c^{\text{fwd}} \leftarrow \frac{R}{n_r}\psi$.
19: Construct an $(n_z + n_r, n_r)$ MDS code for the flow.
20: **else**
21: The problem is infeasible.
22: **end if**

This algorithm is a combination of the algorithms in [21] and [9]. The algorithm in [9] can find the $(n_z + n_r)$ min-cost

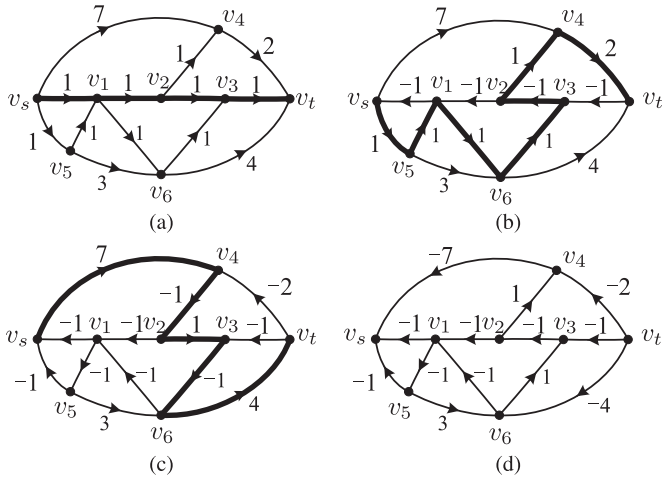


Fig. 7. A directed graph and its iterations; (a) the original graph; (b) the graph after the 1st iteration; (c) the graph after the 2nd iteration; (d) the graph after the 3rd iteration.

disjoint paths for each n_r such that $1 \leq n_z + n_r \leq |\mathcal{P}|$, while the algorithm in [21] can decide the number of paths, (i.e. $(n_z + n_r)$) to allocate the network error correction flow. Specifically, Algorithm 1 contains a loop structure as in Fig. 6. Upon initializing (Line 1), the algorithm enters into a loop (Line 2–15). In the beginning of the i-th iteration ($1 \leq i \leq |\mathcal{P}|$), we have found $(i-1)$ min-cost disjoint paths, and ϕ is the summation of all unit prices in these $(i-1)$ paths. At the same time, $n_r = i - 1 - n_z$.

First, the iteration tries to find a shortest path from v_s to v_t (Note that there may be multiple shortest paths, but we only need to find one of them.) If there are no paths from v_s to v_t, then $i > |\mathcal{P}|$, which breaks the loop (Line 5). In this case, since $n_r = i - 1 - n_z$ and $i = |\mathcal{P}| + 1$, we have $n_r = |\mathcal{P}| - n_z$. Otherwise ($i \leq |\mathcal{P}|$), we set ψ as the summation of the unit prices of the edges on the path that we have just found (Line 7). Now, ψ is the difference between the summation of all unit prices in i min-cost disjoint paths and that in $(i-1)$ min-cost disjoint paths.

Afterward, the iteration compares ϕ and $n_r\psi$. If $\phi \leq n_r\psi$, we only need to allocate the flow on $(i-1)$ disjoint paths. Therefore, it breaks the loop (Line 10). Otherwise, it is more sensible to allocate the flow on i disjoint paths. Then we update ϕ to be the summation of unit prices in i disjoint paths, and set n_r as $i - n_z$ (Line 12).

In order to let the next iteration (i.e., the $(i+1)$-th iteration) find the $(i+1)$ disjoint shortest paths easily, this iteration reverses all edges on the path it has just found. According to [9], using this method, the next iteration can find the $(i+1)$ disjoint shortest paths simply by searching for a shortest path in the modified graph.

After the loop ceases, the algorithm needs to assert whether the problem is feasible. If $n_r > 0$ (Line 16), which is equivalent to $|\mathcal{P}| > 2z$ in DAG, then the problem is feasible. In this case, we allocate the flow on the $(n_r + n_z)$ disjoint paths we have just found, and construct an MDS code therein (Line 17–19).

Example 6: Consider the network in Fig. 7(a), which is modified from [9]. The unit price of each edge is marked alongside. Let $\mathcal{A} = \mathcal{P}(\mathcal{E}, 1)$ (that is, $n_z = 2$). The algorithm runs as follows:

(0) Initialize: $\phi \leftarrow 0$. $n_r \leftarrow (-n_z) = -2$.

(1) Find the shortest path from v_s to v_t in Fig. 7(a): $v_s \xrightarrow{1} v_1 \xrightarrow{1} v_2 \xrightarrow{1} v_3 \xrightarrow{1} v_t$. The summation of unit prices on this path is $1+1+1+1=4$. Set $\psi \leftarrow 4$. Considering that $\phi = 0$ and $n_r = -2$ currently, $\phi > n_r\psi$. Set $n_r \leftarrow n_r + 1 = -1$, $\phi \leftarrow \phi + \psi = 4$. Inverse the edges on the path just found, resulting in Fig. 7(b).

(2) Find the shortest path in Fig. 7(b): $v_s \xrightarrow{1} v_5 \xrightarrow{1} v_1 \xrightarrow{1} v_6 \xrightarrow{1} v_3 \xrightarrow{-1} v_2 \xrightarrow{1} v_4 \xrightarrow{2} v_t$. The summation of unit prices on this path is $1+1+1+1+(-1)+1+2 = 6$. Set $\psi \leftarrow 6$. Considering that $\phi = 4$ and $\psi = 6$ currently, $\phi > n_r\psi$. Set $n_r \leftarrow n_r + 1 = 0$, $\phi \leftarrow \phi + \psi = 4 + 6 = 10$. Inverse the edges on the path just found, resulting in Fig. 7(c). Note that a negative edge between v_2 and v_3, which was reversed in precede iteration, is reversed back to a positive edge.

(3) Find the shortest path in Fig. 7(c): $v_s \xrightarrow{7} v_4 \xrightarrow{-1} v_2 \xrightarrow{1} v_3 \xrightarrow{-1} v_6 \xrightarrow{4} v_t$. The summation of unit prices on this path is $7+(-1)+1+(-1)+4 = 10$. Set $\psi \leftarrow 10$. Considering $\phi = 11$ and $\psi = 10$, $\phi > n_r\psi$. Set $n_r \leftarrow n_r + 1 = 1$, $\phi \leftarrow \phi + \psi = 10 + 10 = 20$. Inverse the used edges, resulting in Fig. 7(d).

(4) No more paths can be found in Fig. 7(d). The loop ends with $n_r = 1$. The minimum cost three disjoint paths are $v_s \xrightarrow{7} v_4 \xrightarrow{2} v_t$, $v_s \xrightarrow{1} v_1 \xrightarrow{1} v_6 \xrightarrow{1} v_3 \xrightarrow{1} v_t$, and $v_s \xrightarrow{1} v_5 \xrightarrow{1} v_1 \xrightarrow{1} v_6 \xrightarrow{4} v_t$. Allocate $R/n_r = R$ on these paths and employ (3, 1) MDS code.

C. Complexity

Here we analyze the time complexity of this algorithm. The number of iterations in the loop is not greater than $|\mathcal{P}|$, which is less than $|\mathcal{V}|$. The worse time complexity of the shortest path algorithms that can be applied to negative edges, such as Bellman-Ford algorithm [26], [27] and its improved versions [28], is $O(|\mathcal{V}||\mathcal{E}|)$. Therefore, the overall time complexity of the algorithm is $O\left(|\mathcal{V}|^2|\mathcal{E}|\right)$.

D. Optimality

The algorithm can attain the minimum cost among all the route-based flows. For the i-th ($1 \leq i \leq |\mathcal{P}|$) iteration, the algorithm of finding disjoint paths can find the minimal cost allocation on exactly i disjoint paths. The updating value of ψ is actually the increase of summation of the unit prices of the i disjoint paths compared to the summation in the last preceding iteration. Due to the properties of multiple disjoint paths, the increase in the latest iteration is always greater than or equal to all the increase value in preceding iterations (a.k.a. preceding ψs). After the edges on the path just found are reversed, the backward paths in the remaining graph will indicate the minimum cost routing among all possible flows on exactly i disjoint paths. According to Theorem 3 and Lemma 4, combining the algorithm to find multiple shortest paths and the

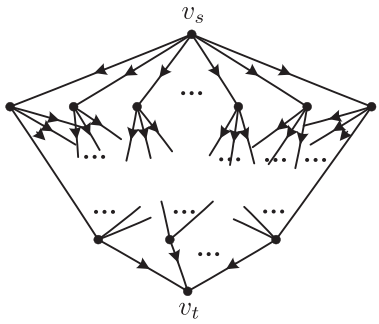


Fig. 8. The topology of the case-study networks.

algorithm to determine the number of paths to use, Algorithm 1 can find the optimal solution among the minimal cost i disjoint paths for $1 \leq i \leq |\mathcal{P}|$.

VIII. Benefit of Intermediate Recoding in DAG

In some cases, the minimum cost network error correction flows do not require recoding at intermediate nodes. In other cases, however, the recoding at intermediate nodes does help. Take the cases in Example 4 for example. Using the algorithm in Section VII-B, we can find that the minimum cost of flows without recoding at intermediate nodes in Example 4(a) and (b) is $6R$ and $9R$ respectively, both of which are greater than the cost of optimal network error correction flows ($\frac{11}{2}R$ and $8R$). Actually, the optimal flows in both examples are not classical network flows, which indicates the necessity of recoding at intermediate nodes.

This section evaluates the benefit brought by recoding at intermediate nodes.

A. Unboundedness of the Benefit

Here is an example to show that the benefit of intermediate recoding can be arbitrarily large compared to non-recoding schemes.

Example 7: Consider a case-study family of flow allocation problems (indexed by $\mathbb{N}$). For the i-th problem ($i \in \mathbb{N}$), the set of possible attack sets is $\mathcal{A} = \mathcal{P}(\mathcal{E}, i)$ and the message rate is $R > 0$. The network is shown in Fig. 8: there are $(i+1)^2$ vertices (denoted as $\mathcal{V}_u$) that are connected with v_s, and the unit price of these connections (denoted as $\mathcal{E}_u$) is $\psi_u = 1$. Simultaneously, there are $(2i+1)$ vertices (denoted as $\mathcal{V}_b$) that connected with v_t, and the unit price of these connections (denoted as $\mathcal{E}_b$) is $\psi_b = 1/(2i+1)$. For any $v_u \in \mathcal{V}_u$ and $v_b \in \mathcal{V}_b$, there is a connection (the set is denoted as $\mathcal{E}_m$) with unit price $\psi_m = 1/\left[(i+1)^2(2i+1)\right]$.

Now we compare the cost of optimal flow on disjoint path c_i^{fwd} and the cost of a feasible flow with intermediate coding c_i^{cod}.

Without Intermediate Recoding: In this DAG,

$$|\mathcal{P}| = \min\{|\mathcal{V}_u|, |\mathcal{V}_b|\} = 2i+1.$$

According to Theorem 3, since $n_z = 2i$ and $n_r = 1$, the best way to allocation disjoint flow is allocating flow equality on $(2i+1)$ disjoint paths, and the flow on each path is R. For this flow, the cost is

$$\begin{aligned} c_i^{\text{fwd}} &= (2i+1)(\psi_u + \psi_m + \psi_b)R \\ &= (2i+1)\left(1 + \frac{1}{(i+1)^2(2i+1)} + \frac{1}{2i+1}\right)R \\ &= \left(2i+2+\frac{1}{(i+1)^2}\right)R. \end{aligned}$$

With Intermediate Recoding: Consider the following network error correction with recoding at the nodes in bottom layer: v_s transmits the message using an $\left((i+1)^2, i^2+1\right)$ MDS code, and nodes in $\mathcal{V}_u$ forward all their received streams to the nodes in $\mathcal{V}_b$. The three vertices in $\mathcal{V}_b$ decode all their received data, cooperate to provide a $(2i+1, 1)$ MDS code, and transmit the $2i+1$ streams to v_t. It is easy to show that any i errors in $\mathcal{E}_u$ or $\mathcal{E}_m$ can be corrected by the vertices in $\mathcal{V}_b$, while v_t can correct any i errors in $\mathcal{E}_b$. Thus, this flow is a feasible flow. Its cost is

$$\begin{aligned} c_i^{\text{cod}} &= \psi_u \frac{(i+1)^2}{i^2+1}R + \psi_m \frac{(i+1)^2(2i+1)}{i^2+1}R + \psi_b(2i+1)R \\ &= \left(\frac{(i+1)^2}{i^2+1} + \frac{1}{i^2+1} + 1\right)R \\ &= \left(2 + \frac{2i+1}{i^2+1}\right)R. \end{aligned}$$

Comparison: The ratio of the minimum cost of network error correction with recoding at intermediate nodes to that without recoding is upper bounded by

$$\gamma_i \leq \frac{c_i^{\text{cod}}}{c_i^{\text{fwd}}} = \frac{\left(2 + \frac{2i+1}{i^2+1}\right)R}{\left(2i+2+\frac{1}{(i+1)^2}\right)R} = \frac{2 + \frac{2i+1}{i^2+1}}{2i+2+\frac{1}{(i+1)^2}}.$$

Note that when $i \to +\infty$, the right hand side of the inequality goes to zero, which results in

$$\gamma_i \to 0 \quad (i \to +\infty).$$

Thus, the benefit of recoding at intermediate nodes can be tremendous.

B. Benefits in Random Graph

The previous subsection shows that recoding at intermediate nodes may bring great benefit in some strong structured instances. Numerical analysis, however, shows that the benefit is quite limited in small random graphs.

Fig. 9 simulates the possible benefit in random graphs where $\mathcal{A} = \mathcal{P}(\mathcal{E}, 1)$. In Fig. 9(a), the x-axis represents the probability of the existing of edge between two vertices, while the y-axis represents the average of the ratio of a lower bound of minimum cost with recoding at intermediate nodes to the minimum cost without recoding. Different lines show the results with distinct vertex numbers. In Fig. 9(b), the x-axis is the same as that in Fig. 9(a), while the y-axis represents the empirical probability

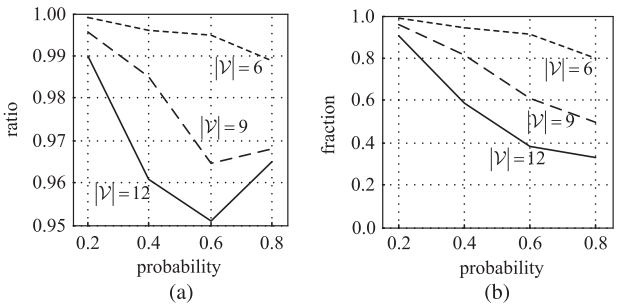


Fig. 9. Benefit of recoding at intermediate nodes in random graphs. (a) Average ratio of the lower bound with intermediate recoding to the minimum cost without recoding. (b) Fraction of cost without intermediate recoding attaining the lower bound.

of the minimum cost without recoding attaining the lower bound with recoding at intermediate nodes. The details of the simulation, including how to generate the networks, how to calculate the lower bound for the ratio, and so on, are provided in Appendix E. The simulation results show that the benefit in random graphs with few vertices is quite limited.

The numerical results show that the benefit in random graphs with few vertices is quite limited. Additionally, codes without intermediate recoding seem to perform better in smaller networks. This phenomenon implies an increasing need of intermediate recoding when the size of network grows.

IX. Conclusion

This paper considered the allocation of network error correction flow to combat Byzantine attacks. We have formulated the flow allocation problem, and found some approaches to allocate the flow in an economical way. Further, we have also investigated the necessity of recoding at intermediate nodes. Unfortunately, we have not come up with a universal solution to find the optimal network error correction flow in general networks and general cost functions. Finding the minimum cost network error correction flow in general is an interesting and nontrivial open problem.

Appendix A
Tradeoff

There were a lot of researches on network error correction flow to combat Byzantine attacks. Previous results showed that a tradeoff occurs when the flow resource is given. The following theorem summarizes the tradeoff.

Theorem 4 (Tradeoff): Given the network $\mathcal{G}=(\mathcal{V},\mathcal{E})$.

(1) For flows $\mathbf{f}_{\mathcal{E}}^{(1)},\mathbf{f}_{\mathcal{E}}^{(2)}\geq\mathbf{0}_{\mathcal{E}}$ such that $\mathbf{f}_{\mathcal{E}}^{(1)}\leq\mathbf{f}_{\mathcal{E}}^{(2)}$ and sets $\mathcal{A}^{(1)},\mathcal{A}^{(2)}\in 2^{\mathcal{E}}$ such that $\mathcal{A}^{(1)}\supseteq\mathcal{A}^{(2)}$,

$$\sup\left\{R:\mathbf{f}_{\mathcal{E}}^{(1)}\in\mathcal{F}_{R^{(1)},\mathcal{A}^{(1)}}\right\}\leq\sup\left\{R:\mathbf{f}_{\mathcal{E}}^{(2)}\in\mathcal{F}_{R^{(2)},\mathcal{A}^{(2)}}\right\};$$

(2) For flows $\mathbf{f}_{\mathcal{E}}^{(1)},\mathbf{f}_{\mathcal{E}}^{(2)}\geq\mathbf{0}_{\mathcal{E}}$ such that $\mathbf{f}_{\mathcal{E}}^{(1)}\leq\mathbf{f}_{\mathcal{E}}^{(2)}$ and rates $R^{(1)},R^{(2)}\geq 0$ such that $R^{(1)}\geq R^{(2)}$,

$$\left\{\mathcal{A}:\mathbf{f}_{\mathcal{E}}^{(1)}\in\mathcal{F}_{R^{(1)},\mathcal{A}}\right\}\subseteq\left\{\mathcal{A}:\mathbf{f}_{\mathcal{E}}^{(2)}\in\mathcal{F}_{R^{(2)},\mathcal{A}}\right\};$$

(3) For sets $\mathcal{A}^{(1)},\mathcal{A}^{(2)}\subsetneqq\mathcal{P}(\mathcal{E})$ such that $\mathcal{A}^{(1)}\subseteq\mathcal{A}^{(2)}$ and rates $R^{(1)},R^{(2)}\geq 0$ such that $R^{(1)}\leq R^{(2)}$,

$$\mathcal{F}_{R^{(1)},\mathcal{A}^{(1)}}\subseteq\mathcal{F}_{R^{(2)},\mathcal{A}^{(2)}}.$$

Previous works mainly focus on the optimization of the first two sets. Since set $\{R:\mathbf{f}_{\mathcal{E}}\in\mathcal{F}_{R,\mathcal{A}}\}$ is a continuous interval starting from zero, numerous researches on finding the supremum of supported rate (such as [3]) actually optimized on the this set. Researches on the error correction ability of network error correction code actually optimized on the set $\{\mathcal{A}:\mathbf{f}_{\mathcal{E}}\in\mathcal{F}_{R,\mathcal{A}}\}$. The minimum cost network error correction flow problem in this paper actually optimizes on the set $\mathcal{F}_{R,\mathcal{A}}$ and tries to minimize the flow on every edge, which can be regarded as a multi-objective optimization. Therefore, we define a function Cost as the criterion to combine the objective functions into a single goal.

Appendix B
Proof of Proposition 1

Homogeneity and unboundedness are obvious so we only prove that the set $\mathcal{F}_{R,\mathcal{A}}$ is convex.

For any $\mathbf{f}_{\mathcal{E}}^{(1)},\mathbf{f}_{\mathcal{E}}^{(1)}\in\mathcal{F}_{R,\mathcal{A}}$, both $\mathbf{f}_{\mathcal{E}}^{(1)}$ and $\mathbf{f}_{\mathcal{E}}^{(2)}$ can support rate R and combat adversary $\mathcal{A}$. Due to 1), for any $\lambda^{(1)},\lambda^{(2)}\in[0,1]$ such that $\lambda^{(1)}+\lambda^{(2)}=1$, flows $\lambda^{(1)}\mathbf{f}_{\mathcal{E}}^{(1)}$ and $\lambda^{(2)}\mathbf{f}_{\mathcal{E}}^{(2)}$ can support rate $\lambda^{(1)}R$ and $\lambda^{(2)}R$ respectively. Consider a network error correction code that concatenates the two codes on flow $\lambda^{(1)}\mathbf{f}_{\mathcal{E}}^{(1)}$ and $\lambda^{(2)}\mathbf{f}_{\mathcal{E}}^{(2)}$, and it will support rate R and combat adversary $\mathcal{A}$. Thus,

$$\lambda^{(1)}\mathbf{f}_{\mathcal{E}}^{(1)}+\lambda^{(2)}\mathbf{f}_{\mathcal{E}}^{(2)}\in\mathcal{F}_{R,\mathcal{A}}.$$

That proves the convexity of the set $\mathcal{F}_{R,\mathcal{A}}$.

Appendix C
Proof of Lemma 3

Proof of Lemma 3: Now we try to construct a spanning tree over graph $\mathcal{G}$ such that

1) the root of the tree is v_s; and
2) none of the edges in the tree belong to $\mathcal{E}_a$.

The method to construct this tree is as follows:

(Step 1) On the one hand, let $\mathcal{V}_s^{(1)}=\{v_s\}$. Obviously, $\left|\mathcal{V}_s^{(1)}\right|=1$. On the other hand, since $\mathcal{E}_c^F\left(\mathcal{V}_s^{(1)}\right)\not\subseteq\mathcal{E}_a$, there exists an edge $e^{(1)}\in\mathcal{E}_c^F\left(\mathcal{V}_s^{(1)}\right)$ such that $e^{(1)}\notin\mathcal{E}_a$. Let $\mathcal{E}^{(1)}=\left\{e^{(1)}\right\}$. Obviously, $\mathcal{E}^{(1)}\bigcap\mathcal{E}_a=\varnothing$.

(Step 2) On the one hand, let $\mathcal{V}_s^{(2)}=\mathcal{V}_s^{(1)}\bigcup\left\{\mathrm{Head}(e^{(1)})\right\}$. Since $e^{(1)}\in\mathcal{E}_c^F\left(\mathcal{V}_s^{(1)}\right)$, $\mathrm{Head}\left(e^{(1)}\right)\notin\mathcal{V}_s^{(1)}$, so $\left|\mathcal{V}_s^{(2)}\right|=2$. On the other hand, since $\mathcal{E}_c^F\left(\mathcal{V}_s^{(2)}\right)\not\subseteq\mathcal{E}_a$, there exists an edge $e^{(2)}\in\mathcal{E}_c^F\left(\mathcal{V}_s^{(2)}\right)$ such that $e^{(2)}\notin\mathcal{E}_a$. Let $\mathcal{E}^{(2)}=\mathcal{E}^{(1)}\bigcup\left\{e^{(2)}\right\}$. Since $\mathcal{E}^{(1)}\bigcap\mathcal{E}_a=\varnothing$ and $e^{(2)}\notin\mathcal{E}_a$, $\mathcal{E}^{(2)}\bigcap\mathcal{E}_a=\varnothing$.

...

(Step i, $1 < i < |\mathcal{V}|$) On the one hand, let $\mathcal{V}_s^{(i)} = \mathcal{V}_s^{(i-1)} \bigcup \left\{\text{Head}\left(e^{(i-1)}\right)\right\}$. Since $e^{(i-1)} \in \mathcal{E}_c^F\left(\mathcal{V}_s^{(i-1)}\right)$, $\text{Head}\left(e^{(i-1)}\right) \notin \mathcal{V}_s^{(i-1)}$, so $\left|\mathcal{V}_s^{(i)}\right| = i$. On the other hand, since $\mathcal{E}_c^F\left(\mathcal{V}_s^{(i)}\right) \not\subseteq \mathcal{E}_a$, there exists an edge $e^{(i)} \in \mathcal{E}_c^F\left(\mathcal{V}_s^{(i)}\right)$ such that $e^{(i)} \notin \mathcal{E}_a$. Let $\mathcal{E}_s^{(i)} = \mathcal{E}_s^{(i-1)} \bigcup \left\{e^{(i)}\right\}$. Since $\mathcal{E}^{(i-1)} \bigcap \mathcal{E}_a = \varnothing$ and $e^{(i)} \notin \mathcal{E}_a$, $\mathcal{E}^{(i)} \bigcap \mathcal{E}_a = \varnothing$.

...

(Step $|\mathcal{V}|$) Let $\mathcal{V}_s^{(|\mathcal{V}|)} = \mathcal{V}_s^{(|\mathcal{V}|-1)} \bigcup \left\{\text{Head}\left(e^{(|\mathcal{V}|-1)}\right)\right\}$. Since $e^{(|\mathcal{V}|-1)} \in \mathcal{E}_c^F\left(\mathcal{V}_s^{(|\mathcal{V}|-1)}\right)$, $\text{Head}\left(e^{(|\mathcal{V}|-1)}\right) \notin \mathcal{V}_s^{(|\mathcal{V}|-1)}$. Therefore, $\left|\mathcal{V}_s^{(|\mathcal{V}|)}\right| = |\mathcal{V}|$, which leads to $\mathcal{V}_s^{(|\mathcal{V}|)} = \mathcal{V}$.

Using these $|\mathcal{V}|$ steps, we have constructed a spanning tree $\mathcal{T} = \left(\mathcal{V}, \mathcal{E}^{(|\mathcal{V}|-1)}\right)$ of $\mathcal{G}$ such that $\mathcal{E}^{(|\mathcal{V}|-1)} \bigcap \mathcal{E}_a = \varnothing$. Since there exists a path (without loop) between arbitrary two vertices in a tree, there exists a path p from v_s to v_t. Moreover, since $\mathcal{E}^{(|\mathcal{V}|-1)} \bigcap \mathcal{E}_a = \varnothing$ and $\mathcal{E}_p \subseteq \mathcal{E}^{(|\mathcal{V}|-1)}$, we have $\mathcal{E}_p \bigcap \mathcal{E}_a = \varnothing$. Therefore, $p \notin \mathcal{P}_{\mathcal{E}_a}$. □

Appendix D
Proof of Theorem 2

Theorem 2, an extension of [3, Theorem 2], is a direct result of Lemma 1, Lemma 5, and Lemma 6.

Lemma 5: Let $R > 0$ and $\mathcal{A} \subseteq 2^{\mathcal{E}}$. Let $\mathcal{E}_c$ be a cut in the graph where the forward edge set is $\mathcal{E}_c^F$ and the backward edge set is $\mathcal{E}_c^B$. $\forall\, \mathcal{E}_a \in \mathcal{A}$,

$$\mathcal{F}_{R,\mathcal{A}} \subseteq \left\{\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}} : \left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \mathcal{E}_a}\right\|_1 \geq R\right\};$$

Proof: Suppose set $\mathcal{E}_a$ is attacked and both transmitter and receiver have known this fact by some methods. In this condition, for any flow $\mathbf{f}_{\mathcal{E}}$, at most $\left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \mathcal{E}_a}\right\|_1$ rate can be transmitted correctly. Thus, for any flow $\mathbf{f}_{\mathcal{E}}$ that needs to support rate R, it should satisfy that $\left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \mathcal{E}_a}\right\|_1 \geq R$. □

Lemma 6: Let $R > 0$ and $\mathcal{A} \subseteq 2^{\mathcal{E}}$. Let $\mathcal{E}_c$ be a cut in the graph where the forward edge set is $\mathcal{E}_c^F$ and the backward edge set is $\mathcal{E}_c^B$. For any $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ such that $\mathcal{E}_c^B \subseteq \mathcal{E}_a^{(1)}$ and $\mathcal{E}_c^B \subseteq \mathcal{E}_a^{(2)}$,

$$\mathcal{F}_{R,\mathcal{A}} \subseteq \left\{\mathbf{f}_{\mathcal{E}} \geq \mathbf{0}_{\mathcal{E}} : \left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}\right\|_1 \geq R\right\}.$$

Proof: Fix $\mathcal{E}_a^{(1)}, \mathcal{E}_a^{(2)} \in \mathcal{A}$ such that $\mathcal{E}_c^B \subseteq \mathcal{E}_a^{(1)}$ and $\mathcal{E}_c^B \subseteq \mathcal{E}_a^{(2)}$. We can prove (by a contradiction) that for any flow $\mathbf{f}_{\mathcal{E}}$ that needs to support rate R, it should satisfy that $\left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}\right\|_1 \geq R$.

Suppose $\left\|\mathbf{f}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}\right\|_1 < R$. According to the Pigeonhole Principle, among all 2^{NR} messages in the message set that can be sent reliably in N channel uses, there exist two distinct message symbols $m^{(1)}$ and $m^{(2)}$ such that the code words on edge set $\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)$ are identical. Let $\mathbf{x}_{\mathcal{E}_c^F}^{(1)} = \left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$ and $\mathbf{x}_{\mathcal{E}_c^F}^{(2)} = \left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$ be the codewords on the cut $\mathcal{E}_c^F$ when $m^{(1)}$ and $m^{(2)}$ are being sent along respectively. When $m^{(1)}$ is transmitted and attacked by $\mathcal{E}_a^{(1)}$, the adversary can change the codeword from $\left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$ to $\left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$. When $m^{(2)}$ is transmitted and attacked by $\mathcal{E}_a^{(2)}$, the adversary can change the codeword from $\left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$ to $\left(\mathbf{x}^{(1)}_{\mathcal{E}_c^F \setminus \left(\mathcal{E}_a^{(1)} \bigcup \mathcal{E}_a^{(2)}\right)}, \mathbf{x}^{(2)}_{\mathcal{E}_a^{(1)}}, \mathbf{x}^{(1)}_{\mathcal{E}_a^{(2)} \setminus \mathcal{E}_a^{(1)}}\right)$. Therefore, the nodes aside the sink node can not distinguish these two messages. That leads to a contraction. □

Proof of Theorem 2: Lemma 5 leads to $\mathcal{F}_{R,\mathcal{A}} \subseteq \mathcal{B}_{R,\mathcal{A}}^{(1)}\left(\mathcal{E}_c^F\right)$, while Lemma 6 leads to $\mathcal{F}_{R,\mathcal{A}} \subseteq \mathcal{B}_{R,\mathcal{A}}^{(2)}\left(\mathcal{E}_c^F, \mathcal{E}_c^B\right)$. Taking Lemma 1 into consideration as well, we have $\mathcal{F}_{R,\mathcal{A}} \subseteq \mathcal{B}_{R,\mathcal{A}}(\mathcal{E}_c)$. □

Appendix E
Details of the Numerical Analysis

This appendix introduces the details of the numerical analysis in Section VIII-B.[3]

Having fixed the vertex number $|\mathcal{V}| > 2$ and the probability of edge existence $p \in (0, 1]$, we generate a large number of instances in the following way [29]: for every instance i to generate, let $\{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$ be the vertices in the network. Let v_1 be the source node and $v_{|\mathcal{V}|}$ be the sink node. For every v_{j_1} and v_{j_2} $(1 \leq j_1 < j_2 \leq |\mathcal{V}|)$, the edge between v_{j_1} and v_{j_2} occurs independently with probability p and the unit price of each edge is independently and uniformly selected from [0,1]. Next, the max-flow algorithm is used to check whether the minimum cardinality of cuts in the instance is greater than $2z = 2$. If so, instance i is a feasible instance, and a lower bound for network error correction flow allowing recoding (denoted as c_i^{rel}) and the minimum cost for flows without recoding at intermediate nodes (denoted as c_i^{fwd}) are calculated by the methods in the Sections VI-A and VII-B respectively. Let $\mathcal{I}(|\mathcal{V}|, p)$ be the set of feasible instances. The average ratio in Fig. 9(a) is defined as

$$\text{ratio}\,(|\mathcal{V}|, p) = \frac{1}{|\mathcal{I}\,(|\mathcal{V}|, p)|} \sum_{i \in \mathcal{I}(|\mathcal{V}|, p)} \frac{c_i^{\text{rel}}}{c_i^{\text{fwd}}}$$

and the fraction in Fig. 9(b) is

$$\text{fraction}\,(|\mathcal{V}|, p) = \frac{\left|\left\{i \in \mathcal{I}\,(|\mathcal{V}|, p) : c_i^{\text{rel}} = c_i^{\text{fwd}}\right\}\right|}{|\mathcal{I}\,(|\mathcal{V}|, p)|}.$$

[3]All related codes, which are developed in MATLAB, can be found at: https://drive.google.com/file/d/0By2m48ItFbTeZTRhY1I5aG9hTDg/.

ACKNOWLEDGMENT

The authors would like to thank Shenghao Yang, Britt Fugitt, Jiang Zhu, Xiangming Meng, and two anonymous reviewers for their helpful comments.

REFERENCES

[1] D. Silva and F. Kschischang, "Adversarial error correction for network coding: models and metrics," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2008, pp. 1246–1253.
[2] S. Kim, T. Ho, M. Effros, and A. Avestimehr, "Network error correction with unequal link capacities," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2009, pp. 1387–1394.
[3] S. Kim, T. Ho, M. Effros, and A. Avestimehr, "Network error correction with unequal link capacities," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1144–1164, Feb. 2011.
[4] T. Ho, S. Kim, Y. Yang, M. Effros, and S. Avestimehr, "On network error correction with limited feedback capacity," in *Proc. Inf. Theory Appl. Workshop*, 2011, pp. 1–3.
[5] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, Jan. 2008.
[6] O. Kosut, L. Tong, and D. Tse, "Nonlinear network coding is necessary to combat general Byzantine attacks," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2009, pp. 593–599.
[7] O. Kosut, L. Tong, and D. Tse, "Polytope codes against adversaries in networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 2423–2427.
[8] O. Kosut, L. Tong, and D. Tse, "Polytope codes against adversaries in networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3308–3344, Jun. 2014.
[9] R. Bhandari, "Optimal physical diversity algorithms and survivable networks," in *Proc. IEEE Symp. Comput. Commun.*, 1997, pp. 433–441.
[10] N. Cai and R. Yeung, "Network coding and error correction," in *Proc. IEEE Inf. Theory Workshop*, Oct. 2002, pp. 119–122.
[11] T. Ho *et al.*, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Oct. 2004, p. 144.
[12] T. Ho *et al.*, "Byzantine modification detection in multicast networks with random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2798–2803, Jun. 2008.
[13] R. W. Yeung and N. Cai, "Network error correction, I: Basic concepts and upper bounds," *Commun. Inf. Syst.*, vol. 6, no. 1, pp. 19–35, 2006.
[14] S. Yang, R. Yeung, and C.-K. Ngai, "Refined coding bounds and code constructions for coherent network error correction," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1409–1424, Mar. 2011.
[15] R. Matsumoto, "Construction algorithm for network error-correcting codes attaining the singleton bound," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 90, no. 9, pp. 1729–1735, Sep. 2007.
[16] X. Guang, F.-W. Fu, and Z. Zhang, "Construction of network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 1030–1047, Feb. 2013.
[17] Y. Yang, T. Ho, and W. Huang, "Network error correction with limited feedback capacity," 2013. [Online]. Available: http://arxiv.org/abs/1312.3823/
[18] D. Lun *et al.*, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2608–2623, Jun. 2006.
[19] T. Cui and T. Ho, "Minimum cost integral network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 2736–2740.
[20] J. Tan and M. Medard, "Secure network coding with a cost criterion," in *Proc. IEEE Int. Symp. Model. Optim. Mobile, Ad Hoc Wireless Netw.*, Apr. 2006, pp. 1–6.
[21] Z. Xiao, Y. Li, and J. Wang, "Allocation of network error correction flow on disjoint paths," *Tsinghua Sci. Technol.*, vol. 20, no. 2, pp. 182–187, Apr. 2015.
[22] Z. Xiao, Y. Li, X. Su, and J. Wang, "Processing delays do not degrade network error-correction capacity in directed networks," *IEEE Commun. Lett.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7093115
[23] W. Huang, M. Langberg, and J. Kliewer, "Connecting multiple-unicast and network error correction: Reduction and unachievability," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2015, pp. 1–6.
[24] R. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 2, pp. 116–118, Apr. 1964.
[25] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2345–2364, Jun. 2006.
[26] R. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, pp. 87–90, 1958.
[27] J. Ford and R. Lester, "Network flow theory," RAND Corp., Santa Monica, CA, USA, Aug. 1956, vol. 16, p. 923.
[28] J. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quart. Appl. Math.*, vol. 27, pp. 526–530, 1970.
[29] E. N. Gilbert, "Random plane networks," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 533–543, 1961.

Zhiqing Xiao received the B.S. degree from Beijing University of Posts and Telecommunications, China, in 2011, and he is currently pursuing the Ph.D. degree in Department of Electronic Engineering, Tsinghua University. His research interests include network error correction, network information-theoretic security, and network information theory.

Yunzhou Li (M'06) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2004. Currently, he is a Professorship Researcher at Tsinghua University. He mainly focuses on signal processing technologies in wireless and mobile communications, including spatial-time signal processing, channel estimation, multi-user detection, and synchronization algorithms for CDMA/OFDM system. He is also interested in analysis, optimization design and enhancement of cellular system and WLAN.

Ming Zhao (M'98) received the B.S. and Ph.D. degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 1993 and 1998, respectively. In 1998, he joined the faculty of Tsinghua University. He is currently a Professorship Researcher at the School of Information Science and Technology, Tsinghua University. He is experienced in the R&D of wireless and mobile communication, and his interests include modulation, channel coding, channel capacity, resource management, and network protocol. He has published over 100 conference and journal papers.

Xibin Xu (M'00) received the B.S. degree from the University of Electronic Science and Technology of China in 1988, and the M.S. degree from Tsinghua University, Beijing, China, in 1992. He joined the faculty of Tsinghua University in 1992 and is currently a Professorship Researcher at the School of Information Science and Technology, Tsinghua University. His research interests are in the area of wireless communications, including transmission and networking technologies of 5G. He has published over 100 conference and journal papers.

Jing Wang (M'99) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1983 and 1986, respectively. He has been on the faculty at Tsinghua University since 1986. He is currently a Professor at the School of Information Science and Technology, Tsinghua University. He serves as the Vice Director of the Tsinghua National Lab for Information Science and Technology. His research interests are in the area of wireless communications, including transmission and networking technologies of 5G. He has published more than 150 conference and journal papers.