Robust Network Coding in the Presence of Untrusted Nodes

Da Wang, Danilo Silva, *Member, IEEE*, and Frank R. Kschischang, *Fellow, IEEE*

Abstract—While network coding can be an efficient means of information dissemination in networks, it is highly susceptible to "pollution attacks," as the injection of even a single erroneous packet has the potential to corrupt each and every packet received by a given destination. Even when suitable error-control coding is applied, an adversary can, in many interesting practical situations, overwhelm the error-correcting capability of the code. To limit the power of potential adversaries, a broadcast transformation is introduced, in which nodes are limited to just a single (broadcast) transmission per generation. Under this broadcast transformation, the multicast capacity of a network is changed (in general reduced) from the number of edge-disjoint paths between source and sink to the number of internally disjoint paths. Exploiting this fact, a family of networks is proposed whose capacity is largely unaffected by a broadcast transformation. This results in a significant achievable transmission rate for such networks, even in the presence of adversaries.

Index Terms—Adversarial nodes, broadcast transformation, error correction, JLC networks, multicast capacity, network coding.

I. INTRODUCTION

NETWORK coding [1] is a promising approach for efficient information dissemination in packet networks. Network coding generalizes routing, allowing nodes in the network not only to switch packets from input ports to output ports, but also to combine incoming packets in some manner to form outgoing packets. For example, in *linear* network coding, fixed-length packets are regarded as vectors over a finite field $\mathbb{F}_q$, and nodes in the network form outgoing packets as $\mathbb{F}_q$-linear combinations of incoming packets. For the single-source multi-cast problem, it is known that linear network coding suffices to achieve the network capacity [2], [3].

Recently, the problem of error correction in network coding has received significant attention due to the fact that pollution attacks can be catastrophic. Indeed, the injection of even a single erroneous packet somewhere in the network has the potential to corrupt each and every packet received by a given sink node. This problem was first investigated from an edge-centric perspective [4], where a number of packet errors could arise in any of the links in the network. Alternatively, under a node-centric perspective, it is assumed that an adversarial node may join the network and transmit corrupt packets on all its outgoing links, but the other links in the network remain free of error.

One approach, investigated in [5] and [6], for dealing with the pollution problem is to apply cryptographic techniques to ensure the validity of received packets, permitting corrupted packets to be discarded by each node, and therefore preventing the contamination of other packets. This approach typically requires the use of large field and packet sizes, which leads to computationally expensive operations at the nodes and possibly to significant transmission delay. These requirements may be acceptable in the large-file-downloading scenario, but may be incompatible with delay-constrained applications such as streaming-media distribution.

Another approach (and the one followed in this paper) is to look for end-to-end coding techniques that require little or no intelligence at the internal nodes. Jaggi *et al.* [7] show that, if C is the network capacity (per transmission-generation) and z is the min-cut from the adversary to a destination, then a rate of $C - 2z$ packets per generation is achievable. The same rate can also be achieved using the subspace approach introduced by Kötter and Kschischang [8], [9]. A higher rate $C - z$ can be achieved using a scheme proposed in [7] (see also [10]) if the source and sink nodes are allowed to share a secret (i.e., if they have common information not available to the adversary).

In all of the end-to-end techniques mentioned above, we observe that the min-cut from the adversary to a sink node has a significant impact on the achievable rates. If z is large—for instance, if $z = C$—then the adversary can jam the network with no hope of recovery. It is important, therefore, to conceive of protocols that induce per-generation network topologies that can perform well, even in the presence of adversaries.

The central question of this paper is the following:

What simple changes to a protocol (and, hence, to the induced network topology) might be effective in reducing the influence of an adversary, while not (greatly) affecting the rate of reliable communication?

We show that in some important special cases it is indeed possible to constrict potential adversaries, without any sacrifice of network capacity.

D. Wang was with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada. He is now with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: dawang@mit.edu).

D. Silva was with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada. He is now with the School of Electrical and Computer Engineering, State University of Campinas, Campinas, SP 13083-970, Brazil (e-mail: danilo@decom.fee.unicamp.br).

F. R. Kschischang is with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada (e-mail: frank@comm.utoronto.ca).

In this paper, we introduce the concept of a *broadcast transformation*, which essentially constrains potential adversaries to sending the same packet on all its outgoing links. In the case of a single malicious node, this effectively enforces $z = 1$. In order for such a transformation to be possible, we introduce the concept of a *trusted node* that performs the role of broadcasting traffic. A beneficial side-effect of a broadcast transformation is to lower the encoding complexity, since each node only needs to compute a single outgoing packet in each round of communication.

In practice, such a broadcasting feature could be implemented at trusted network gateways. For example, in overlay network applications, it could be implemented by ISPs at their gateways, through the use of deep packet inspection or similar technologies. Note that the broadcast constraint is effectively enforced if all packets in the same generation[1] from the same user have identical payload (although with different headers corresponding to different destination addresses). Thus, for each user/generation pair, the network gateway could simply store the payload of the first packet it receives and drop any subsequent packets that have different payloads (while also flagging such a user as "suspicious"). It is worth mentioning that, for wireless networks, this constraint is automatically satisfied due to the broadcast nature of wireless communication [12], so the results of this paper are also naturally applicable in this case.

In general, a broadcast transformation can reduce capacity (significantly, in some cases), unless the network has special connectivity properties. We will show that the maximum number of *internally disjoint paths* between source and sink, rather than edge-disjoint paths, becomes the key parameter. This result implies that robustness to node failures and robustness to adversarial attacks are closely related concepts. We then examine a class of networks, which we call *d-diverse* networks, that have excellent robustness properties. This class of networks is strongly inspired by the work of Jain *et al.* in [13] on robust and scalable network topologies. We show that, under certain conditions, no loss in capacity is incurred when performing broadcast conversion in such *d*-diverse networks.

The remainder of this paper is organized as follows. In Section II, we review some basic concepts of graph theory and network coding. In Section III, we introduce an adversarial model for communication over untrusted networks. In Section IV, we introduce the broadcast transformation and characterize the achievable rates of broadcast-constrained networks by relating it to parameters of the original network. In Section V, we introduce *d*-diverse networks and study their robustness properties. In Section VI, we present our conclusions.

II. PRELIMINARIES

A. Graph Theory

In this paper, a *graph* always means a directed multigraph, i.e., all edges are directed and multiple edges between nodes[2] are allowed. If $\mathcal{G}$ is a graph, then $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$ denote its vertex set

---

[1]Here, we assume the use of generation-based network coding, as proposed in [11].

[2]We will use "vertex" and "node" interchangeably in this paper.

and edge set, respectively. Let $\mathbb{Z}_+ = \{1, 2, 3, \ldots\}$. We assume that $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{G}) \times \mathbb{Z}_+$, where the third component is used to distinguish among multiple edges between the same nodes.

For $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}(\mathcal{G})$, let $[\mathcal{A}, \mathcal{B}]$ denote the set of all edges (a, b, i) in $\mathcal{G}$ such that $a \in \mathcal{A}$ and $b \in \mathcal{B}$. We may also denote $[a, \mathcal{B}] \triangleq [\{a\}, \mathcal{B}], [\mathcal{A}, b] \triangleq [\mathcal{A}, \{b\}]$ and $[a, b] \triangleq [\{a\}, \{b\}]$. For $[\mathcal{A}, \mathcal{B}]$ and any other concept that implicitly depends on $\mathcal{G}$, we will use a subscript such as $[\mathcal{A}, \mathcal{B}]_{\mathcal{G}}$ if the graph is not clear from the context.

If $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, then $\mathcal{G} - \mathcal{S}$ is the graph consisting of the vertex set $\mathcal{V}(\mathcal{G}) \setminus \mathcal{S}$ and edge set $\mathcal{E}(\mathcal{G}) \setminus [\mathcal{V}, \mathcal{S}] \cup [\mathcal{S}, \mathcal{V}]$.

Let $|\mathcal{S}|$ denote the cardinality of a set $\mathcal{S}$. For nodes u and v, if $\|[u, v]\| > 0$, then u is called a *parent* of v, while v is called a *child* of u. We use $\Gamma^-(v)$ and $\Gamma^+(v)$ to denote, respectively, the set of all parents and the set of all children of a node v.

Let $\mathsf{indegree}(v) = \|[\mathcal{V}(\mathcal{G}), v]\|$ and $\mathsf{outdegree}(v) = \|[v, \mathcal{V}(\mathcal{G})]\|$.

For $e \in [u, v]$, let $\mathsf{tail}(e) = u$ and $\mathsf{head}(e) = v$. Also, for $\mathcal{E} \subseteq \mathcal{E}(\mathcal{G})$, let $\mathsf{tail}(\mathcal{E}) \triangleq \cup_{e \in \mathcal{E}} \mathsf{tail}(e)$ and, similarly, let $\mathsf{head}(\mathcal{E}) \triangleq \cup_{e \in \mathcal{E}} \mathsf{head}(e)$.

For $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, let $\bar{\mathcal{S}} \triangleq \mathcal{V}(\mathcal{G}) \setminus \mathcal{S}$. For distinct nodes s and t, if $s \in \mathcal{S}$ and $t \in \bar{\mathcal{S}}$, then $[\mathcal{S}, \bar{\mathcal{S}}]$ is called an *s,t-edge cut*. Let

$$\mathsf{mincut}(s, t) \triangleq \min_{\substack{\mathcal{S} \subseteq \mathcal{V}(\mathcal{G}): \\ s \in \mathcal{S} \not\ni t}} \|[\mathcal{S}, \bar{\mathcal{S}}]\|$$

denote the minimum size of an s, t-edge cut. Note that $\mathsf{mincut}(s, t)$ is often denoted by $\kappa'(s, t)$. For convenience, define also

$$\mathsf{mincut}(\mathcal{A}, t) \triangleq \min_{\substack{\mathcal{S} \subseteq \mathcal{V}(\mathcal{G}): \\ \mathcal{A} \subseteq \mathcal{S} \not\ni t}} \|[\mathcal{S}, \bar{\mathcal{S}}]\|.$$

A *path* is a sequence of vertices such that from each vertex there is an edge to the next vertex in the sequence. The first and last vertices in a finite path are called *end vertices*, and the other vertices are called *internal vertices*.

For distinct nodes s and t, a set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G}) \setminus \{s, t\}$ is called an *s, t-vertex cut* if $\mathcal{G} - \mathcal{S}$ has no path connecting s and t. Note that for an s, t-vertex cut to exist, t cannot be a child of s. In that condition, let $\kappa(s, t)$ denote the minimum size of an s, t-vertex cut.

Two paths are called *edge-disjoint* if they have no edges in common, and are called *internally disjoint* if they have no internal nodes in common. Let $\lambda'(s, t)$ denote the maximum number of pairwise edge-disjoint paths from a node s to a node t and let $\lambda(s, t)$ denote the maximum number of pairwise internally disjoint paths from s to t.

We will frequently refer to the edge and vertex versions of Menger's Theorem on directed graphs [14] (the former is also known as the Max-flow Min-cut Theorem).

Theorem 1 (Menger's Theorem, Edge Version): For any vertices s and t, $\lambda'(s, t) = \mathsf{mincut}(s, t)$.

Theorem 2 (Menger's Theorem, Vertex Version): For any vertices s and t, if $\|[s, t]\| = 0$, then $\lambda(s, t) = \kappa(s, t)$.

B. Network Coding

A *(single-source) multicast network* $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$ consists of a (directed multi)graph $\mathcal{G}$ with a distinguished *source node* s and a set of *sink nodes* $\mathcal{T} \not\ni s$. In a multicast problem, each sink node requests the same message that is observed at the source node.

Each link in the network is assumed to transport, free of errors, a packet of a certain fixed size. A packet in a link entering a node is said to be an incoming packet to that node, and similarly a packet in a link leaving a node is said to be an outgoing packet from that node.

When network coding is used, the source node produces each of its outgoing packets as an arbitrary function of the message it observes. Also, each nonsource node produces each of its outgoing packets as an arbitrary function of its incoming packets. The set of functions applied by all nodes in the network specifies a *network code*. If each sink node, by observing its incoming packets, is able to correctly identify the source message, then we say that the decoding is successful.

Let q be the size of the set from which packets are selected and let Ω be the set from which the source message is selected. The *rate* of communication is defined as

$$R(\Omega, q) \triangleq \log_q |\Omega|$$

which is the amount of information, measured in packets, that can be conveyed by the source message.

A rate R is said to be *achievable* for a network $\mathcal{N}$ if, for any $\epsilon > 0$, there exist q and Ω with $R(\Omega, q) \geq R$, along with a corresponding network code, such that the probability of unsuccessful decoding is smaller than ϵ.

For a multicast network $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$, define

$$C(\mathcal{N}) \triangleq \min_{t \in \mathcal{T}} \mathsf{mincut}_{\mathcal{G}}(s, t).$$

A key result in [1] is that a rate R is achievable for $\mathcal{N}$ if and only if

$$R \leq C(\mathcal{N}).$$

For this reason, $C(\mathcal{N})$ is referred to as the *capacity* of a multicast network $\mathcal{N}$.

III. UNTRUSTED MULTICAST NETWORKS

In this section, we describe a node-centric adversarial model for networks that can be subject to pollution attacks. This model will be used in the remainder of the paper for the computation of achievable rates.

We start with the definition of an untrusted multicast network. Consider a multicast network. A node is said to be *trusted* if it is guaranteed to behave according to a specified network coding protocol; otherwise, it is said to be *untrusted*. In particular, a trusted node cannot be controlled by an adversary, while an untrusted node may (or may not) be so. An *untrusted multicast network* $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ is a multicast network $(\mathcal{G}, s, \mathcal{T})$ with a specified set of untrusted nodes $\mathcal{U} \subseteq \mathcal{V}(\mathcal{G}) \setminus \{s\}$ such that all nodes in $\mathcal{V}(\mathcal{G}) \setminus \mathcal{U}$ are trusted.

An adversarial model for communication over an untrusted multicast network may be specified as follows. The adversary chooses a set of adversarial nodes $\mathcal{A} \subseteq \mathcal{U}$ with $|\mathcal{A}| \leq w$ prior to the beginning of the session. The set $\mathcal{A}$ is unknown to source and sink nodes, but remains fixed during the whole session. The adversary controls the nodes in $\mathcal{A}$, which are allowed to transmit any arbitrary packets on their outgoing links and also to cooperate with each other. Since an adversarial node cannot be counted as a sink node, we say that decoding is successful if each node in $\mathcal{T} \setminus \mathcal{A}$ can correctly recover the source message.

Several end-to-end error control schemes have been proposed to ensure reliable communication over an untrusted network [7]–[10], [15]. The rates achievable by these schemes depend on further assumptions on the system model. In this paper, we focus on the two most basic of these models. The *omniscient adversary* (OA) model refers to the case where no constraints are imposed on the knowledge or computational power of the adversary. If an additional assumption is made that common randomness is available between the source and sink nodes, then resulting scenario is called the *shared secret* (SS) model.

Achievable rates under these models are often stated from an edge-centric perspective, i.e., assuming that the adversary controls a certain number of edges. Below we restate these results from a node-centric perspective.

Theorem 3 ([7], [9]): Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network with at most w adversarial nodes. Under the shared secret model, it is possible to achieve the rate

$$R^{\mathsf{OA}}(\mathcal{N}, w) \triangleq \min_{\substack{\mathcal{A} \subseteq \mathcal{U}: \\ |\mathcal{A}| \leq w}} \min_{t \in \mathcal{T} \setminus \mathcal{A}} R^{\mathsf{OA}}(s, t, \mathcal{A}) \qquad (1)$$

where

$$R^{\mathsf{OA}}(s, t, \mathcal{A}) \triangleq [\mathsf{mincut}(s, t) - 2\mathsf{mincut}(\mathcal{A}, t)]^+.$$

Theorem 4 ([7], [10], [15]): Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network with at most w adversarial nodes. Under the omniscient adversary model, it is possible to achieve the rate

$$R^{\mathsf{SS}}(\mathcal{N}, w) \triangleq \min_{\substack{\mathcal{A} \subseteq \mathcal{U}: \\ |\mathcal{A}| \leq w}} \min_{t \in \mathcal{T} \setminus \mathcal{A}} R^{\mathsf{SS}}(s, t, \mathcal{A}) \qquad (2)$$

where

$$R^{\mathsf{SS}}(s, t, \mathcal{A}) \triangleq [\mathsf{mincut}(s, t) - \mathsf{mincut}(\mathcal{A}, t)]^+.$$

We will use (1) and (2) as benchmarks to evaluate the effective throughput of a multicast network in the presence of adversaries.

Note that when there is no adversary, both expressions reduce to the capacity of the underlying network, i.e.,

$$R^{\mathsf{OA}}(\mathcal{N}, 0) = R^{\mathsf{SS}}(\mathcal{N}, 0) = C(\mathcal{N}).$$

From Theorems 3 and 4, we observe that, for an adversarial set $\mathcal{A}$ and a sink node t, the quantity $\mathsf{mincut}(\mathcal{A}, t)$ can have a severe impact on the achievable rate of the untrusted network. If $\mathsf{mincut}(\mathcal{A}, t)$ is large compared to $\mathsf{mincut}(s, t)$, then the adversary can overwhelm the system with corrupt packets, preventing successful decoding.
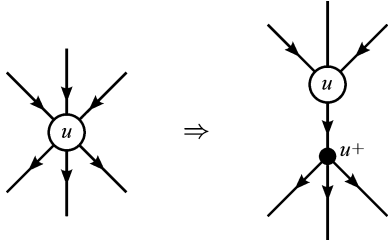
Fig. 1. Broadcast transformation.

IV. BROADCAST TRANSFORMATION

In this section, we propose an approach to restrict the min-cut between adversarial nodes and sink nodes, which can lead to potentially higher achievable rates over untrusted networks. The idea is to force each adversarial node to transmit only copies of the same packet, effectively constraining its outdegree to be at most 1. As we do not know beforehand which nodes are adversarial, the constraint must be enforced on every *untrusted* node. This operation can be represented graphically by introducing a new node u^+, as described in Fig. 1. Here, u^+ is a *trusted node* that only replicates the packet received. The overall operation, which we refer to as a *broadcast transformation*, is formally defined below.

Definition 1: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The *broadcast transformation* of $\mathcal{N}$, denoted by $\beta(\mathcal{N})$, is an untrusted multicast network $(\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U})$ with $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ given by

$$\hat{\mathcal{V}} = \mathcal{V} \cup \{u^+ : u \in \mathcal{U}\}$$
$$\hat{\mathcal{E}} = (\mathcal{E} \setminus [\mathcal{U}, \mathcal{V}]) \cup \{(u, u^+, 1) : u \in \mathcal{U}\} \cup [\mathcal{U}, \mathcal{V}]^+$$

where $[\mathcal{U}, \mathcal{V}]^+ = \{(u^+, v, i) : (u, v, i) \in [\mathcal{U}, \mathcal{V}]\}$.

After a broadcast transformation, adversarial nodes can only do limited harm, as shown in the following simple result.

Proposition 5: Let $\beta(\mathcal{N})$ be the broadcast transformation of an untrusted multicast network $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$. For $0 \leq w \leq C(\beta(\mathcal{N}))$, we have

$$R^{\mathsf{OA}}(\beta(\mathcal{N}), w) \geq [C(\beta(\mathcal{N})) - 2w]^+$$
$$R^{\mathsf{SS}}(\beta(\mathcal{N}), w) \geq [C(\beta(\mathcal{N})) - w]^+$$

with equality if $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$.

Proof: Let $(\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U}) = \beta(\mathcal{N})$. The pair of inequalities follows immediately from Definition 1 and Theorems 3 and 4 by noticing that $\mathsf{mincut}_{\hat{\mathcal{G}}}(\mathcal{A}, t) \leq |\mathcal{A}|$ for any $\mathcal{A} \subseteq \mathcal{U}$ and any $t \in \mathcal{T} \setminus \mathcal{A}$.

For the case $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$, let $t \in \mathcal{T}$ be any node satisfying $\mathsf{mincut}_{\hat{\mathcal{G}}}(s, t) = C(\beta(\mathcal{N}))$. Note that t must have at least $C(\beta(\mathcal{N}))$ distinct parents in $\mathcal{G}$, all of which are untrusted. Take any w of such parents to form a set $\mathcal{A}$. Then $\mathsf{mincut}_{\hat{\mathcal{G}}}(\mathcal{A}, t) = w$, which shows that both inequalities can be met with equality. ∎

In general, applying a broadcast transformation may reduce $C(\beta(\mathcal{N}))$, the multicast capacity of the resulting network. Still, the reduction in the jamming capability of the adversary may compensate for this loss and yield a higher achievable rate. This

tradeoff, which is captured by Proposition 5, will be shown to be indeed favorable in certain meaningful situations. More specifically, we are interested in studying networks for which $C(\beta(\mathcal{N}))$ is equal or approximately equal to $C(\mathcal{N})$. If this is the case, we will say that $\mathcal{N}$ is a *robust* network.

In the remainder of the paper, we restrict attention to the case $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$, where all nonsource nodes are untrusted. This case not only has analytical advantages, but also seems to be the case of most practical relevance.

For a multicast network $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$, define

$$\Lambda(\mathcal{N}) \triangleq \min_{t \in \mathcal{T}} \lambda_{\mathcal{G}}(s, t).$$

The following theorem shows that the multicast capacity of a broadcast-transformed network has a nice graph-theoretical characterization in terms of the original network.

Theorem 6: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network with $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$. Then

$$C(\beta(\mathcal{N})) = \Lambda(\mathcal{N}).$$

Proof: The proof is closely related to the standard argument used to derive Theorem 2 from the Max-flow Min-cut Theorem.

Let $\beta(\mathcal{N}) = (\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U})$. Since $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$, the broadcast transformation replaces each nonsource node by a node followed by an edge followed by a node, as illustrated in Fig. 1. Thus, if two paths in $\mathcal{G}$ are internally disjoint, then they will also be internally (and, therefore, edge-) disjoint in $\hat{\mathcal{G}}$. Conversely, if two paths in $\mathcal{G}$ are not internally disjoint, i.e., they share a node v, then they will also share the two nodes v and v^+ and the edge $(v, v^+, 1)$ in $\hat{\mathcal{G}}$ and, therefore, will not be edge-disjoint in $\hat{\mathcal{G}}$. Thus, for any $t \in \mathcal{T}$, the maximum number of internally disjoint paths from s to t in $\mathcal{G}$ must be equal to the maximum number of edge-disjoint paths from s to t in $\hat{\mathcal{G}}$, i.e., $\lambda_{\mathcal{G}}(s, t) = \lambda'_{\hat{\mathcal{G}}}(s, t) = \mathsf{mincut}_{\hat{\mathcal{G}}}(s, t)$. The result now follows from the definitions of $\Lambda(\mathcal{N})$ and $C(\beta(\mathcal{N}))$. ∎

We now give some examples of robust and nonrobust networks.

Example 1: Consider the network $\mathcal{N}$ in Fig. 2, where s is the source node and all other nodes $v_1, \ldots, v_9$ are untrusted sink nodes. Note that, for any v_i, we have $\mathsf{mincut}(s, v_i) = 3$ and, therefore, $C(\mathcal{N}) = 3$. Meanwhile, $\lambda(s, v_5) = 1$, so $C(\beta(\mathcal{N})) = \Lambda(\mathcal{N}) = 1$. Thus, $\mathcal{N}$ is not a robust network. ∎

Example 2: To make the network in Fig. 2 robust, we can increase the diversity of internally disjoint paths to v_5 and v_6 by letting v_5 and v_6 have multiple parents. This may result in a network $\mathcal{N}$ as shown in Fig. 3. Now, for all i, we have $\mathsf{mincut}(s, v_i) = 3$ and $\lambda(s, v_i) = 3$. Thus, $C(\mathcal{N}) = 3$ and $C(\beta(\mathcal{N})) = \Lambda(\mathcal{N}) = 3$. Therefore, $\mathcal{N}$ is a robust network. ∎

V. d-DIVERSE NETWORKS

In this section, we study a special class of networks, which we call *d-diverse networks*, that have simultaneously good capacity and robustness properties. This class of networks is motivated by the notion of parent diversity illustrated in Example 2.
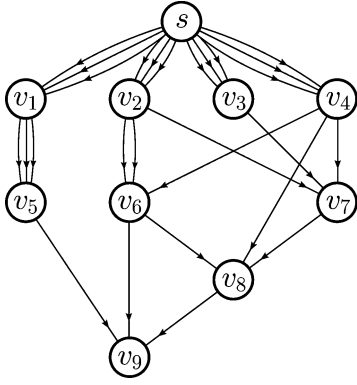
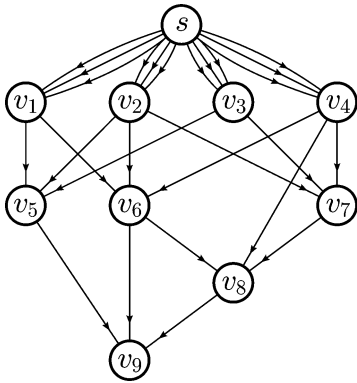Fig. 2.   Nonrobust network with $C(\mathcal{N}) = 3$ and $C(\beta(\mathcal{N})) = 1$.

Fig. 3.   Robust network with $C(\beta(\mathcal{N})) = C(\mathcal{N}) = 3$.

Definition 2: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$ be an acyclic multicast network. The *(parent) diversity* of a nonsource node $v \in \mathcal{V}(\mathcal{G}) \backslash \{s\}$ is defined as

$$d(v) \triangleq |\Gamma^-(v) \backslash \{s\}| + |[s, v]|.$$

The *(parent) diversity* of $\mathcal{N}$ is defined as

$$d(\mathcal{N}) \triangleq \min_{v \in \mathcal{V}(\mathcal{G}) \backslash \{s\}} d(v).$$

If $d(\mathcal{N}) = d$, then $\mathcal{N}$ is called a *d-diverse network*.

For any node that is nonadjacent to the source node, the parent diversity is exactly the cardinality of its parent set. For a node that is adjacent to the source node, this interpretation remains true if we replace each edge coming from the source node by an edge followed by a node followed by an edge. This slight twist in the definition is required due to the special role that a source node has in a network problem.

The following is the main result of this section.

Theorem 7: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$ be an acyclic network. Then

$$\Lambda(\mathcal{N}) \geq d(\mathcal{N}).$$

In particular, if $\mathsf{indegree}(t) = d(\mathcal{N})$ for some $t \in \mathcal{T}$, then

$$C(\mathcal{N}) = \Lambda(\mathcal{N}) = d(\mathcal{N}).$$

Theorem 7 shows that, for large enough d, a d-diverse network not only has good multicast capacity but is also robust. In particular, when designing a network, one might focus solely on achieving high parent diversity, obtaining good capacity and robustness as natural consequences. It is important to note that, while $C(\mathcal{N})$ and $\Lambda(\mathcal{N})$ are global parameters of the network, the diversity $d(\mathcal{N})$ (or rather $d(v)$ for each node v) is a parameter that depends only on local information available at a node. Therefore, it should be relatively easy to construct a d-diverse network by enforcing $d(v) \geq d$ at each node. This is indeed the case for the class of JLC networks, as discussed later in Example 3.

In order to prove Theorem 7, we start with a lemma that characterizes minimal vertex cuts in a graph.

Lemma 8: Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nonadjacent nodes s and t. Then every minimal s, t-vertex cut is given by $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$ for some s, t-edge cut $[\mathcal{S}, \bar{\mathcal{S}}]$. In particular

$$\lambda_{\mathcal{G}}(s, t) = \min_{[\mathcal{S}, \bar{\mathcal{S}}]} |\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])| \qquad (3)$$

where the minimization is taken over all s, t-edge cuts $[\mathcal{S}, \bar{\mathcal{S}}]$ such that $s \notin \mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$.

Proof: First, note that if $[\mathcal{S}, \bar{\mathcal{S}}]$ is an s, t-edge cut such that $s \notin \mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$, then $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$ is indeed an s, t-vertex cut. This follows from the fact that removing $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$ from $\mathcal{G}$ also removes all the edges in $[\mathcal{S}, \bar{\mathcal{S}}]$.

We now show that if $\mathcal{A}$ is a minimal s, t-vertex cut, then there exists some s, t-edge cut $[\mathcal{S}, \bar{\mathcal{S}}]$ with $s \notin \mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$ such that $\mathcal{A} = \mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$. For this, consider the graph $\mathcal{G} - \mathcal{A}$. Since $\mathcal{A}$ is an s, t-vertex cut, the graph $\mathcal{G} - \mathcal{A}$ has two components. Let $\mathcal{A}_s$ and $\mathcal{A}_t$ be the components that contain s and t, respectively. Let $\mathcal{S} = \mathcal{A}_s \cup \mathcal{A}$; then $\bar{\mathcal{S}} = \mathcal{A}_t$. Note that $[\mathcal{S}, \bar{\mathcal{S}}]$ is an s, t-edge cut. Moreover, $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}]) \subseteq \mathcal{A}$, otherwise $\mathcal{A}$ would not separate s and t. Since $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$ is also an s, t-vertex cut and $\mathcal{A}$ is minimal, we conclude that $\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}]) = \mathcal{A}$. In addition, we must have $[\mathcal{S}, \bar{\mathcal{S}}] \cap [s, \mathcal{V}] = \emptyset$, otherwise $s \in \mathcal{A}$, which is impossible by the definition of an s, t-vertex cut.

Now the result follows immediately from Theorem 2. ∎

We can now give a proof of Theorem 7.

Proof of Theorem 7: Let $t \in \mathcal{T}$. First, suppose t is not adjacent to s. Let $[\mathcal{S}, \bar{\mathcal{S}}]$ be some s, t-edge cut achieving the minimization in (3). Since the graph $\mathcal{G}$ is directed acyclic, it has at least one topological ordering. Let u be the first node in $\bar{\mathcal{S}}$ according to some topological ordering, i.e., $u \in \bar{\mathcal{S}}$ is a node whose parents are all in $\mathcal{S}$. We have

$$\begin{aligned} \lambda_{\mathcal{G}}(s, t) &= |\mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])| \\ &\geq |\Gamma^-(u)| \\ &\geq d(\mathcal{N}) \end{aligned} \qquad (4)$$

where (4) follows from the fact that $|[s, u]| = 0$, since $s \notin \mathsf{tail}([\mathcal{S}, \bar{\mathcal{S}}])$.

Now, suppose t is adjacent to s. Let $m = |[s, t]|$. Consider a new network $\mathcal{N}' = (\mathcal{G}', s, \mathcal{T})$, where $\mathcal{G}' = \mathcal{G} - [s, t]$. Note
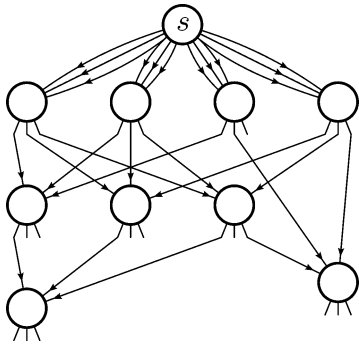
Fig. 4. d-diverse JLC network with $k = 12$ and $d = 3$.

that $d(\mathcal{N}') \geq d(\mathcal{N}) - m$. Using the argument above on $\mathcal{N}'$, we obtain that

$$\lambda_{\mathcal{G}'}(s,t) \geq d(\mathcal{N}') \geq d(\mathcal{N}) - m.$$

Returning to the original network, we have

$$\lambda_{\mathcal{G}}(s,t) = \lambda_{\mathcal{G}'}(s,t) + m \geq d(\mathcal{N}).$$

From the above arguments, it follows that $\Lambda(\mathcal{N}) \geq d(\mathcal{N})$. The special case follows immediately since $\Lambda(\mathcal{N}) \leq C(\mathcal{N}) \leq$ indegree(t), for all $t \in \mathcal{T}$. ∎

As an application of Theorem 7, consider the case of a network in which all nonsource nodes are sink nodes with diversity exactly d, and such that there are no parallel edges between nodes, except possibly emanating from the source node. Then the multicast capacities both before and after broadcast transformation are exactly equal to d. Note that, as the indegree of any nonsource node is exactly d, any removed edge would result in a smaller capacity. Thus, we may conclude that, given a fixed number of edges, the network capacity is maximized by having nodes select incoming edges from distinct parents rather than from the same parent. This result holds even if all nonsource nodes are untrusted, provided a broadcast transformation is performed.

Example 3 (JLC Networks): We now describe a class of networks that has not only good theoretical properties but also potential for practical applications. The protocol for constructing and operating these networks has been proposed by Jain *et al.* [13] as a scalable and robust solution to peer-to-peer data dissemination with network coding. We refer to any network constructed according to their protocol as a *JLC network*.

An example of a JLC network is depicted in Fig. 4. The network is acyclic, and all nonsource nodes are sinks. Initially, the network contains only the source node (or server), which has k (potential) outgoing links. Here, each link represents a stream of unit bandwidth. At any time, the server maintains a list of k available links for download. When a new node joins the network, it requests from the server d download links. The server randomly picks d links from the pool of available links, and updates its list with d potential links originating from the new

node. Therefore, the network always has k links (i.e., streams of unit bandwidth) available for download.

It is easy to ensure that a JLC network is d-diverse by performing a simple protocol modification. When a new node joins the network, rather than choosing the d upstream links completely at random from the k available links (thereby allowing the possibility of fewer than d distinct parents), the server simply needs to provide the new node with d links from d distinct parents. Note that, in practice, $k \gg d^2$, so the k available links come from at least $l = \lceil k/d \rceil \gg d$ parents. Hence, the modification can be done easily. ∎

VI. CONCLUSION

We have introduced the broadcast transformation of a network, which restricts the influence of potential adversaries by limiting them to a single transmission opportunity per generation. For networks with a sufficient diversity of internally disjoint paths from source to sink(s), the multicast capacity may not be greatly affected by this transformation. In particular, for a class of networks called d-diverse networks, the full capacity is maintained when d is sufficiently large. Combined with error control for network coding, the proposed approach may be an effective means of dealing with adversaries, particularly in application scenarios such as real-time media streaming, where alternative (e.g., cryptographic) methods may be cost-prohibitive.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
[2] 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.
[3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
[4] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. 2002 IEEE Inform. Theory Workshop*, Bangalore, India, Oct. 20–25, 2002, pp. 119–122.
[5] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proc. 40th Annual Conf. Inform. Sciences and Systems*, Princeton, NJ, Mar. 2006, pp. 857–863.
[6] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jul. 24–29, 2007, pp. 556–560.
[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.
[8] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.
[9] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.
[10] L. Nutman and M. Langberg, "Adversarial models and resilient schemes for network coding," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, ON, Canada, Jul. 6–11, 2008, pp. 171–175.
[11] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 2003, pp. 40–49.

[12] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 789–804, Mar. 2006.

[13] K. Jain, L. Lovász, and P. A. Chou, "Building scalable and robust peer-to-peer overlay networks for broadcasting using network coding," *Distrib. Comput.*, vol. 19, no. 4, pp. 301–311, 2007.

[14] D. B. West, *Introduction to Graph Theory*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2001.

[15] D. Silva and F. R. Kschischang, "A key-based error control scheme for network coding," in *Proc. 11th Canadian Workshop Inform. Theory*, Ottawa, ON, Canada, May 13–15, 2009, pp. 5–8.

Da Wang received the B.A.Sc. (Hons.) degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2008, and the M.S. degree in the Department of Electrical Engineering and Computer Science (EECS), Massachusetts Institute of Technology (MIT), Cambridge, in 2010. He is currently pursuing the Ph.D. degree in EEC at MIT.

His research interests lie in the areas of communication and information theory.

Danilo Silva (S'06–M'09) received the B.Sc. degree from the Federal University of Pernambuco, Recife, Brazil, in 2002, the M.Sc. degree from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil, in 2005, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 2009, all in electrical engineering.

In 2009, he was a Postdoctoral Fellow at the University of Toronto and at the Ecole Polytechnique Fédérale de Lausanne (EPFL). Since 2010, he has been a Postdoctoral Fellow at the School of Electrical and Computer Engineering, State University of Campinas (Unicamp), Brazil. His research interests include channel coding, information theory, and network coding.

Dr. Silva was a recipient of a CAPES Ph.D. Scholarship in 2005 and of the Shahid U.H. Qureshi Memorial Scholarship in 2009.

Frank R. Kschischang (S'83–M'91–SM'00–F'06) received the B.A.Sc. degree (with honors) from the University of British Columbia, Vancouver, BC, Canada, in 1985, and the M.A.Sc. and Ph.D. degrees from the University of Toronto, ON, Toronto, ON, Canada, in 1988 and 1991, respectively, all in electrical engineering.

He is a Professor of Electrical and Computer Engineering at the University of Toronto, where he has been a faculty member since 1991. During 1997–1998, he was a visiting scientist at the Massachusetts Institute of Technology, Cambridge, and in 2005, he was a visiting professor at the ETH, Zurich. His research interests are focused primarily in the area of channel coding techniques, applied to wireline, wireless, and optical communication systems and networks.

Prof. Kschischang was a recipient of the Ontario Premier's Excellence Research Award in 1999, and in 2001 (renewed in 2008) he was awarded the Tier I Canada Research Chair in Communication Algorithms at the University of Toronto. In 2010, he was awarded the Killam Research Fellowship by the Canada Council for the Arts. Jointly with R. Koetter, he received the 2010 Communications Society and Information Theory Society Joint Paper Award. He is a Fellow of IEEE and of the Engineering Institute of Canada. During 1997-2000, he served as an Associate Editor for Coding Theory for the TRANSACTIONS ON INFORMATION THEORY. He also served as technical program co-chair for the 2004 IEEE International Symposium on Information Theory (ISIT), Chicago, and as general co-chair for ISIT 2008, Toronto. He serves as the 2010 President of the IEEE Information Theory Society.