Constricting the Adversary: A Broadcast Transformation for Network Coding

Da Wang, Danilo Silva and Frank R. Kschischang

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-mode 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. In some interesting cases (in particular, in a class of networks introduced by Jain, Lovász and Chou), the network capacity is maintained in broadcast mode. This results in a significant achievable transmission rate for such networks, even in the presence of adversaries.**

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 network coding operations are linear with respect to $\mathbb{F}_q$, i.e., nodes in the network form outgoing packets as $\mathbb{F}_q$-linear combinations of incoming packets. For the single-source multicast 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], [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 results of [8] show that, using the subspace approach introduced in [9], it is possible (in some cases) to achieve a slightly larger rate, upper-bounded by $C - z$. The rate $C - z$ can be achieved using a scheme proposed in [7] if the source and sink nodes are allowed to share a secret (i.e., 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 graph 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.

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. In practice, such a broadcasting feature could be implemented, e.g., at a trusted network gateway.

In general, our proposed broadcast transformation can, in some cases, significantly reduce capacity, unless the network has special connectivity properties. We will show that

The authors are with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: da.wang@utoronto.ca, danilo@comm.utoronto.ca, frank@comm.utoronto.ca).

the maximum number of *internally-disjoint paths* between source and sink, rather than edge-disjoint paths, becomes the key parameter. We specifically examine a class of networks that have been proposed and extensively analyzed by Jain, Lovász and Chou (JLC) in [10]. We show that, under fairly general conditions, no loss in capacity is incurred when performing broadcast conversion in such JLC networks.

The remainder of this paper is organized as follows. In Sec. II we review some basic definitions in network coding. In Sec. III we introduce our adversarial model for communication over untrusted networks along with some examples. In Sec. IV we introduce the broadcast transformation and prove our main result concerning the achievable rates for JLC networks. In Sec. V we present some simulation results focused on practical scenarios and in Sec. VI we present our conclusions.

II. Preliminaries

Let $\mathcal{G}$ be a directed multigraph with vertex set $\mathcal{V}(\mathcal{G})$ and edge set $\mathcal{E}(\mathcal{G})$. We will 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 edges in $\mathcal{G}$ directed from some vertex in $\mathcal{A}$ to some vertex in $\mathcal{B}$. Let $\mathsf{indeg}(v)$ and $\mathsf{outdeg}(v)$ denote the indegree and outdegree, respectively, of a vertex v. Also, define

$$\begin{aligned}\mathsf{mincut}_{\mathcal{G}}(s,t) &\triangleq \min_{s \in \mathcal{A} \subseteq \mathcal{V}(\mathcal{G}) \setminus \{t\}} |[\mathcal{A}, \mathcal{V}(\mathcal{G}) \setminus \mathcal{A}]| \\ \mathsf{mincut}_{\mathcal{G}}(\mathcal{S},t) &\triangleq \min_{\mathcal{S} \subseteq \mathcal{A} \subseteq \mathcal{V}(\mathcal{G}) \setminus \{t\}} |[\mathcal{A}, \mathcal{V}(\mathcal{G}) \setminus \mathcal{A}]|.\end{aligned}$$

We will often omit the subscript $\mathcal{G}$ when the graph is clear from context.

A *(single-source) multicast network* $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$ consists of a directed multigraph $\mathcal{G}$ with a distinguished source node s, which observes a certain message, and a set of sink nodes $\mathcal{T} \not\ni s$, where each node in $\mathcal{T}$ demands the message observed at s.

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 non-source node produces each of its outgoing packets as an arbitrary function of its incoming packets. Each sink node then attempts to recover the source message from its incoming packets. We say that decoding is successful when correct recovery occurs for all sink nodes.

The set of all functions applied by all nodes in the network specifies a *network code*.

Let the packets in the network each consist of M symbols from a finite field $\mathbb{F}_q$ and let Ω denote the codebook from which the source message is selected. The *rate* of Ω is defined as

$$R(\Omega) \triangleq \frac{1}{M} \log_q |\Omega|.$$

A rate R is said to be *achievable* for a network $\mathcal{N}$ if there exists a sequence of codes Ω_i with $R(\Omega_i) \geq R$, along with corresponding network codes, such that the probability of unsuccessful decoding becomes arbitrarily small as $i \to \infty$ (here, q and M are allowed to grow with i).

Define

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

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

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

For this reason, $C(\mathcal{N})$ is usually regarded as the *capacity* of a multicast network $\mathcal{N}$. As shown in [2], [3], this multicast capacity is achievable with linear network coding.

III. Untrusted Multicast Networks

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

Definition 1: An *untrusted multicast network* $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ consists of a multicast network $(\mathcal{G}, s, \mathcal{T})$ together with a set of *untrusted nodes* $\mathcal{U} \subseteq \mathcal{V}(\mathcal{G}) \setminus \{s\}$. The nodes in $\mathcal{V}(\mathcal{G}) \setminus \mathcal{U}$ are called *trusted nodes*.

Our adversarial model for communication over an untrusted multicast network is the following. 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. We say that decoding is successful if each sink node $t \in \mathcal{T} \setminus \mathcal{A}$ can correctly recover the source message.

Let us focus on a specific sink node t and a specific set of adversarial nodes $\mathcal{A} \not\ni t$. In [7], Jaggi *et al.* analyze a similar model where the adversary has the capability to obtain the source message (say, by eavesdropping a sufficient number of packets) prior to sending its own corrupt packets. Note that this model is compatible with ours since we impose no constraint on the eavesdropping capability of the adversary. In such a scenario, it is shown in [7] that the rate

$$\mathsf{mincut}(s,t) - 2\,\mathsf{mincut}(\mathcal{A},t)$$

is achievable.

Results in [8] show that, using subspace codes and a bounded-distance decoder [9], it is also possible to achieve a slightly higher rate, namely

$$R^{\mathsf{BD}}(s,t,\mathcal{A}) \triangleq \mathsf{mincut}(\{s\} \cup \mathcal{A}, t) - 2\,\mathsf{mincut}(\mathcal{A},t).$$

Thus, for the general case of an untrusted network $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ with at most w adversarial nodes, the rate

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

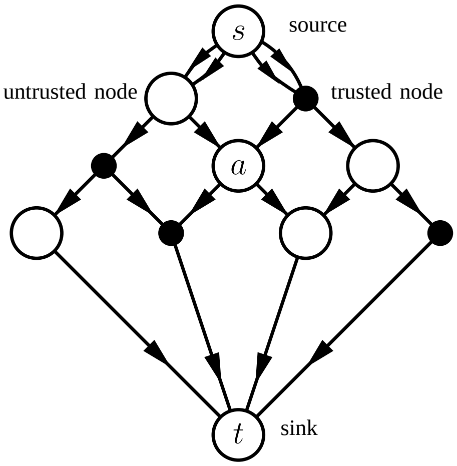

Fig. 1. Untrusted multicast network, with $R^{\mathsf{BD}}(\mathcal{N},1) = 0$ and $R^{\mathsf{SS}}(\mathcal{N},1) = 2$.

is achievable.

As shown in [7], if an additional assumption is made that the source node can share a (small) secret with each of the sink nodes, then it is possible to achieve the rate

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

where

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

We will use (1) as our benchmark to evaluate the robustness of a multicast network in the presence of adversaries, but (2) may sometimes also be used. Note that, since

$$\mathsf{mincut}(\{s\}\cup\mathcal{A},t) \leq \mathsf{mincut}(s,t) + \mathsf{mincut}(\mathcal{A},t)$$

we have $R^{\mathsf{BD}}(s,t,\mathcal{A}) \leq R^{\mathsf{SS}}(s,t,\mathcal{A})$, so a network that performs well under the measure (1) will also perform well under (2).

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

$$R^{\mathsf{BD}}(\mathcal{N},0) = R^{\mathsf{SS}}(\mathcal{N},0) = C(\mathcal{G},s,\mathcal{T}).$$

Example 1: Let $\mathcal{N}$ denote the untrusted multicast network of Fig. 1. There is a single sink node t, and the trusted nodes are the source node s and all nodes represented by a filled circle. By inspection, we find that $\mathsf{mincut}(\{s,a\},t) = \mathsf{mincut}(s,t) = 4$, while $\mathsf{mincut}(a,t) = 2$. It is easy to see that $R^{\mathsf{BD}}(\mathcal{N},1) = 0$ and $R^{\mathsf{SS}}(\mathcal{N},1) = 2$.

We now consider a specific class of network topologies proposed by Jain, Lovász and Chou [10] for its advantages in terms of scalability and robustness to node failures in peer-to-peer applications. Under the protocol proposed in [10], it is possible to practically maintain the network capacity even after nodes join, leave or fail.

Definition 2: A multicast network $\mathcal{N}$ is a *JLC(d,k) network* if it consists only of a source node s, or if it is formed by adjoining to some JLC(d,k) network a sink node t and d edges such that the following properties are satisfied:

(P1) $\mathsf{indeg}(t) = d$;
(P2) each pair of edges entering t that do not come from s must come from distinct nodes;
(P3) $\mathsf{outdeg}(v) \leq d$ for all $v \neq s$;
(P4) $\mathsf{outdeg}(s) \leq k$.

An *untrusted JLC network* is a JLC network where all non-source nodes are untrusted.

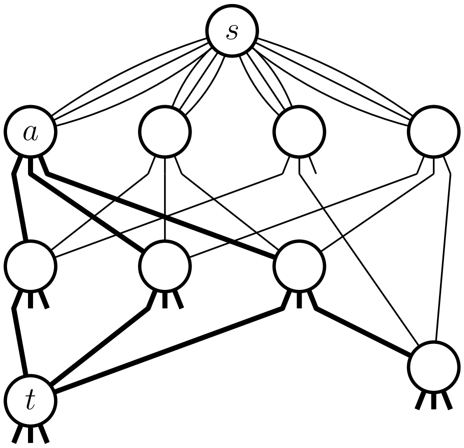

Fig. 2. JLC network with $d = 3$ and $k = 4$.

By construction, a JLC network is an acyclic network where each non-source node t is an (untrusted) sink node with exactly $d - |[s,t]|$ non-source parents. It is easy to see that $\mathsf{mincut}(s,t) = d$ for all $t \neq s$.

Remark 1: The network in Definition 2 is in fact a slight variation of the network proposed in [10], obtained by enforcing property (P2). In [10], edges entering t are randomly selected from nodes whose outdegree has not yet been saturated (i.e., from the pool of potential edges) and therefore it is possible for two of such selected edges to come from the same node. The reason for including (P2) will be clear from Lemma 2 in Section IV. In practice, the edges entering t can still be chosen randomly as long as (P2) is satisfied.

Example 2: An example of a JLC network $\mathcal{N}$ with $d = 3$ and $k = 4$ is shown in Fig. 2. The sink nodes were adjoined in succession from left to right and top to bottom. Suppose a is an adversarial node. Since $\mathsf{mincut}(a,t) = 3 = \mathsf{indeg}(t)$, we obtain that $R^{\mathsf{BD}}(\mathcal{N}) \leq 0$ and $R^{\mathsf{SS}}(\mathcal{N}) = 0$.

From the example above, we observe that the quantity $\mathsf{mincut}(a,t)$ can have a severe impact on the achievable rate for an untrusted multicast network. If $\mathsf{mincut}(a,t)$ is large compared to $\mathsf{mincut}(\{s,a\},t)$, as in the case of a JLC network, then the adversary can overwhelm the system with corrupt packets, preventing successful decoding. In the next section, we explore ways to limit the strength of the adversary without sacrificing network capacity.

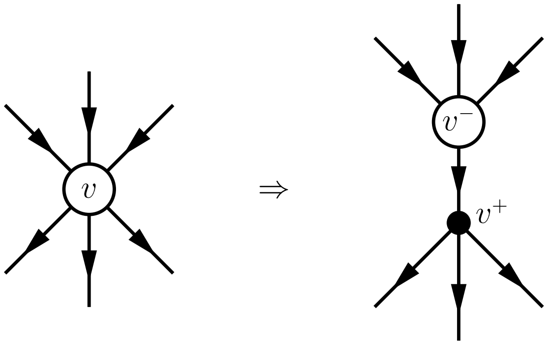

Fig. 3. Broadcast transformation.

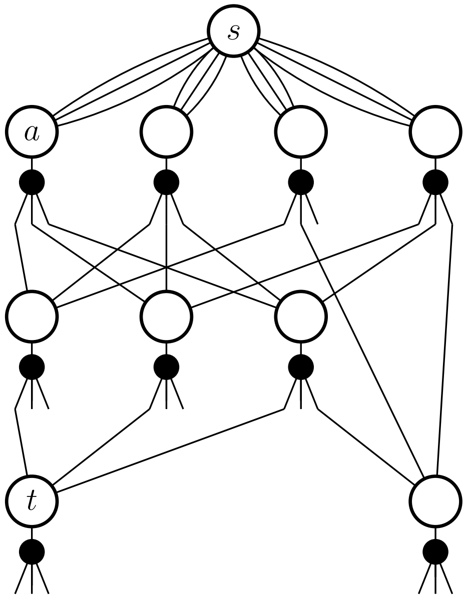

Fig. 4. JLC network with $d = 3$ and $k = 4$ after broadcast transformation.

IV. Network Transformations

We begin by illustrating our approach with an example. Consider again the network in Fig. 2. We see that $\mathsf{mincut}(a, t) = 3$ only because a can inject three distinct packets, which will end up overwriting all packets received by t.

Suppose, however, that we constrain each untrusted node u to send only copies of the same packet. This can be represented graphically by introducing a new node u^+, as described in Fig. 3. Here, u^+ is a *trusted node* that only replicates the packet received. Clearly, we now have $\mathsf{mincut}(a, t) = 1$ in the network of Fig. 4. However, it is not at all obvious that enforcing this constraint on every untrusted node will not severely reduce the network capacity.

While it is clear that, after such a transformation, $\mathsf{mincut}(s, t)$ may be reduced in general, the reduction in $\mathsf{mincut}(a, t)$ may (or may not) compensate for this loss and yield a higher achievable rate. A smooth tradeoff between the two quantities may be achieved by considering a general transformation that limits the outdegree of each untrusted node to at most r.

Definition 3: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For $\mathcal{L} \subseteq \mathcal{E}$, let $\mathcal{L}^+ = \{(u^+, v, i) \,:\, (u, v, i) \in \mathcal{L}\}$. The *degree-$r$ transformation* of $\mathcal{N}$ is an untrusted multicast network $\hat{\mathcal{N}} = (\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U})$, where $\hat{\mathcal{G}}$ is given by

$$\begin{aligned} \mathcal{V}(\hat{\mathcal{G}}) &= \mathcal{V} \cup \{u^+ \,:\, u \in \mathcal{U}\} \\ \mathcal{E}(\hat{\mathcal{G}}) &= [\mathcal{V} \setminus \mathcal{U}, \mathcal{V}] \cup [\mathcal{U}, \mathcal{V}]^+ \cup \bigcup_{u \in \mathcal{U}} \bigcup_{1 \le i \le r} \{(u, u^+, i)\}. \end{aligned}$$

A degree-1 transformation of $\mathcal{N}$ will be called a *broadcast transformation,* denoted by $\beta(\mathcal{N})$.

It is immediate from Definition 3 that if $\hat{\mathcal{N}}$ is a degree-r transformation of $\mathcal{N}$, then for any adversarial set $\mathcal{A}$ in $\hat{\mathcal{N}}$ we have $\mathsf{mincut}(\mathcal{A}, t) \le r|\mathcal{A}|$.

In the remainder of the paper, we treat only the case $r = 1$. This case is interesting not only because it provides the maximum constraint on $\mathsf{mincut}(\mathcal{A}, t)$, but also because it allows useful graph-theoretic tools to be applied in this context.

Let $\lambda'_{\mathcal{G}}(s, t)$ denote the number of edge-disjoint paths from a node s to a node t in $\mathcal{G}$ and let $\lambda_{\mathcal{G}}(s, t)$ denote the number of internally-disjoint paths from s to t in $\mathcal{G}$. The following proposition is part of a standard argument used in graph theory to derive the vertex version of Menger's Theorem from the Max-Flow Min-Cut Theorem [11]. We include its proof for completeness.

Proposition 1: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be an untrusted multicast network with $\mathcal{U} = \mathcal{V}(\mathcal{G}) \setminus \{s\}$, and let $\hat{\mathcal{N}} = (\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U})$ be a broadcast transformation of $\mathcal{N}$. Then $\lambda'_{\hat{\mathcal{G}}}(s, t) = \lambda_{\mathcal{G}}(s, t)$ for all $t \in \mathcal{T}$.

Proof: 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 vertex v, then they will also share the two vertices v and v^+ and the edge (v, v^+) in $\hat{\mathcal{G}}$ and therefore will not be edge-disjoint in $\hat{\mathcal{G}}$. Thus, the maximum number of internally-disjoint paths in $\hat{\mathcal{G}}$ must be equal to the maximum number of edge-disjoint paths in $\mathcal{G}$. ■

The following lemma characterizes internally-disjoint paths in a JLC network.

Lemma 2: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T})$ be a JLC(d, k) network. Then $\lambda_{\mathcal{G}}(s, v) = d$ for all $v \in \mathcal{V}(\mathcal{G}) \setminus \{s\}$.

Proof: (By induction on the size of a JLC(d, k) network)

If $\mathcal{V}(\mathcal{G}) = \{s\}$, we have nothing to prove. Assume that $\mathcal{N}$ is obtained by adjoining a node t to some JLC(d, k) network $\bar{\mathcal{N}} = (\mathcal{G} - t, s, \mathcal{T} \setminus \{t\})$ satisfying $\lambda_{\mathcal{G}-t}(s, v) = d$ for all $v \in \mathcal{V}(\mathcal{G}) \setminus \{s, t\}$. Let $\mathcal{X}$ be the set of non-source parents of t, and recall that $|\mathcal{X}| = d - |[s, t]|$. Let $\mathcal{G}^* = \mathcal{G} - [s, t]$. If t is disconnected from s in $\mathcal{G}^*$ (i.e., if $|\mathcal{X}| = 0$), then it is trivial that $\lambda_{\mathcal{G}}(s, t) = d$, so assume $|\mathcal{X}| \ge 1$.

Suppose that $\lambda_{\mathcal{G}}(s, t) < d$. Then $\lambda_{\mathcal{G}^*}(s, t) < d - |[s, t]|$. By Menger's theorem, there exists a vertex set $\mathcal{A}$ with $|\mathcal{A}| = \lambda_{\mathcal{G}^*}(s, t)$ whose deletion makes t unreachable from s in $\mathcal{G}^*$. Since $|\mathcal{A}| < d - |[s, t]| = |\mathcal{X}|$, there exists at least one $x \in \mathcal{X}$ such that $x \notin \mathcal{A}$. But since t is reachable from x, the deletion of $\mathcal{A}$ must also make x unreachable from s in $\mathcal{G}^*$. By Menger's theorem, $\lambda_{\mathcal{G}^*}(s,x) \leq |\mathcal{A}| < d - |[s,t]|$, which implies $\lambda_{\mathcal{G}}(s,x) < d$ and $\lambda_{\mathcal{G}-t}(s,x) < d$. But this contradicts the assumption, so we must have $\lambda_{\mathcal{G}}(s,t) \geq d$. Observing the $\mathsf{indeg}(t) = d$, we have $\lambda_{\mathcal{G}}(s,t) = d$, which, together with the induction hypothesis, implies that $\lambda_{\mathcal{G}}(s,v) = d$ for all $v \in \mathcal{V}(\mathcal{G}) \setminus \{s\}$. ■

Using Proposition 1 and Lemma 2, we can now compute the achievable rates for a broadcast-constrained JLC network.

Theorem 3: Let $\hat{\mathcal{N}}$ be the broadcast transformation of an untrusted JLC(d,k) network. For $0 \leq w \leq d$, we have

$$R^{\mathsf{BD}}(\hat{\mathcal{N}}, w) = d - 2w$$

$$R^{\mathsf{SS}}(\hat{\mathcal{N}}, w) = d - w.$$

Proof: Let $\mathcal{N} = (\mathcal{G}, s, \mathcal{T}, \mathcal{U})$ be a JLC network such that $\hat{\mathcal{N}} = (\hat{\mathcal{G}}, s, \mathcal{T}, \mathcal{U}) = \beta(\mathcal{N})$. Using Lemma 2, Proposition 1 and the Max-Flow Min-Cut theorem [11], we have

$$\mathsf{mincut}_{\hat{\mathcal{G}}}(s,t) = \lambda'_{\hat{\mathcal{G}}}(s,t) = \lambda_{\mathcal{G}}(s,t) = d, \quad \forall t \in \mathcal{T}.$$

Observe that $d = \mathsf{mincut}_{\hat{\mathcal{G}}}(s,t) \leq \mathsf{mincut}_{\hat{\mathcal{G}}}(\{s\} \cup \mathcal{A}, t) \leq \mathsf{indeg}(t) = d$. Moreover, $\mathsf{mincut}_{\hat{\mathcal{G}}}(\mathcal{A}, t) \leq |\mathcal{A}|$ for all $\mathcal{A}$, and $\mathsf{mincut}_{\hat{\mathcal{G}}}(\mathcal{A}, t) = |\mathcal{A}|$ if $\mathcal{A}$ is a subset of the set of parents of t. The result now follows by applying the definitions (1) and (2). ■

Theorem 3 shows that the broadcast transformation of a JLC network does not result in a decrease in multicast capacity. Moreover, the loss in achievable rate due to the presence of adversaries is limited. Thus, a broadcast transformation of the network of Example 1 results in a nonzero achievable rate, even in the presence of an adversary with $w = 1$.

V. Simulation Results

While Theorem 3 provides a strong theoretical result for an interesting class of networks, not all interesting networks are so easily characterized. In this section we resort to simulation to further investigate the reduction of multicast capacity of certain random networks with the broadcast constraint. In our simulations, we choose $\mathcal{T} = \mathcal{V} \setminus \{s\}$ for all multicast problems. Data points in all graphs are the average of at least 40 trials.

A. Impatient JLC Networks

In an *impatient network*, nodes transmit their outgoing packets before receiving all their incoming packets. The *patience* parameter α, $0 < \alpha \leq 1$, is the fraction of incoming packets that are used to compute the outgoing packets. Although such networks may have lower capacities then their patient counterparts, setting $\alpha < 1$ may be desirable, as the transmission delay from source to destination can potentially be reduced.

In order to compute the multicast capacity of an impatient JLC network, we use the following procedure. Starting from a network with a single (source) node, we iteratively adjoin new nodes according to Definition 2. Each time a node t is adjoined, the min-cut to this node is computed. Then, $(1-\alpha)d$ of its incoming edges are randomly selected and "blocked", i.e., such edges are assigned capacity 0 for subsequent computations.

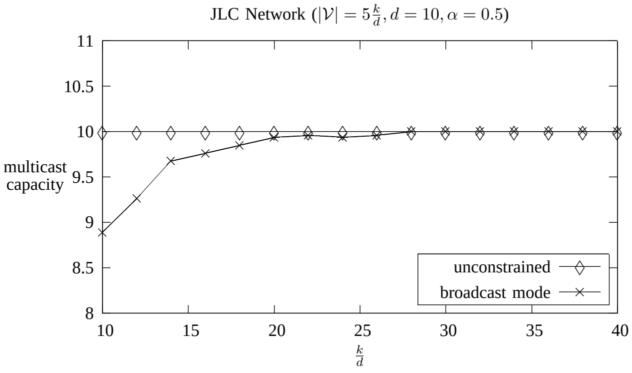

Fig. 5. Capacity of impatient JLC networks as a function of k/d.

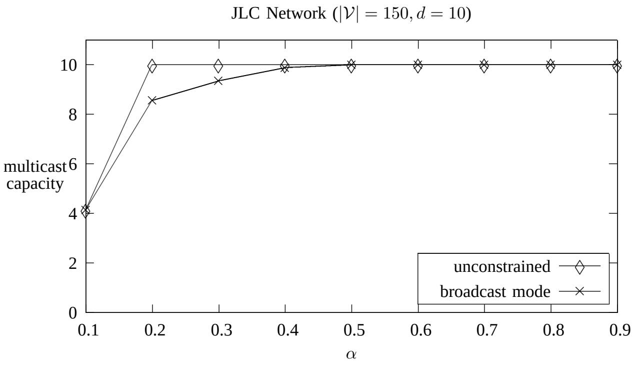

Fig. 6. Capacity of impatient JLC networks as a function of α.

Our simulation results show that impatience does not affect the multicast capacity significantly. For an impatient JLC network with patience $\alpha = 0.5$, $d = 10$, Fig. 5 shows that as the ratio of source outdegree to sink indegree k/d increases, the multicast capacity gradually approaches its upper limit, regardless of the impatient behavior of nodes. Specifically, when $k/d = 15$, the loss of multicast capacity in broadcast mode is less than 5%, and when $k/d \geq 30$, there is essentially no loss of capacity.

Fig. 6 shows the change of multicast capacity of an impatient JLC network as α changes. Naturally, the smaller the α, the less multicast capacity the network has, but only when $\alpha \leq 0.2$ does the multicast capacity suffer severely. Therefore, we can choose $\alpha \geq 0.3$ to achieve shorter transmission delay, without sacrificing much multicast capacity.

B. Complete Graph with Random Edge Capacities

It is also of interest to consider the class of complete graphs with random edge capacities. We use the following construction: in a complete graph with bidirectional edges, we randomly select a source node and, for each edge, we assign its capacity according to a certain probability distribution. Specifically, we used Bernoulli and geometric

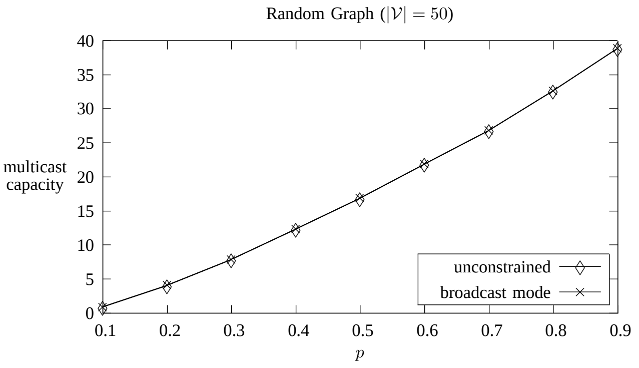

Fig. 7. Complete network with Bernoulli distributed edge capacities.

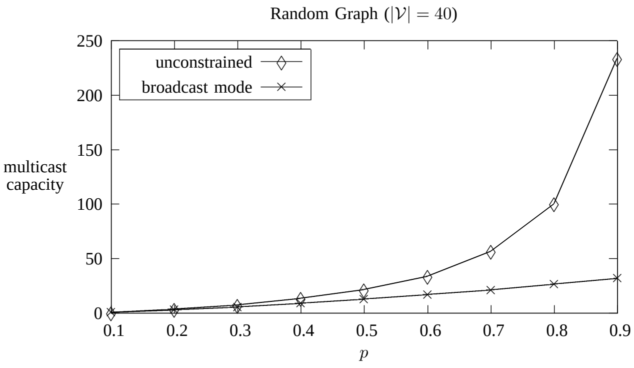

Fig. 8. Complete network with geometric distributed edge capacities.

distributions in our simulation studies. Note that an edge with capacity $c > 1$ can be modeled as c multiple parallel edges, each with capacity 1.

When the network has Bernoulli distributed edge capacities with parameter p, Fig. 7 shows that the multicast capacity of a network under broadcast constraint is essentially the same as that of its unconstrained counterpart.

For a network where each edge has a geometric distributed capacity c, with $\mathsf{Pr}[c = i] = (1-p)p^i$, $i = 0, 1, \ldots$, the performance gap between the unconstrained and the broadcast constrained cases increases with p, as shown in Fig. 8. The reason is that the broadcast transformation effectively limits the outgoing edge capacity of a node to 1, so an increase in the incoming edge capacity does not improve the multicast capacity. By contrast, the multicast capacity in the unconstrained case increases rapidly because both incoming and outgoing edge capacities increase as p increases.

Unlike the JLC networks, it is not clear that this class of complete graphs with random edge capacities corresponds to any realistic networks.

VI. Conclusions

We have introduced the broadcast-mode transformation of a network, which restricts the influence of potential adversaries by limiting them to a single transmission opportunity per generation. In some networks, for example the JLC 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. Combined with error-control coding, this 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. Li, and R. Yeung, "Network information flow," *IEEE Trans. on Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Inform. 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. on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] N. Cai and R. Yeung, "Network coding and error correction," in *Proc. 2002 IEEE Inform. Theory Workshop*, 20-25 Oct. 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*, 24–29 July 2007, pp. 556–560.

[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proc. 26th IEEE Int. Conf. on Computer Commun. (INFOCOM 2007)*, Anchorage, AK, May 2007, pp. 616–624.

[8] D. Silva and F. R. Kschischang, "Achievables rates for network coding in the presence of adversaries," in preparation, 2007.

[9] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, 24–29 June 2007, pp. 791–795.

[10] K. Jain, L. Lovász, and P. A. Chou, "Building scalable and robust peer-to-peer overlay networks for broadcasting using network coding," in *ACM Symp. on Principles of Dist. Computing*, Las Vegas, NV, July 2005, pp. 51–59.

[11] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2001.