Network Reduction for Coded Multiple-Hop Networks

Jinfeng Du†‡, Naomi Sweeting†*, David C. Adams‡ and Muriel Médard†
†Research Lab of Electronics, Massachusetts Institute of Technology, Cambridge, 02139, MA
‡ACCESS Linnaeus Center, Royal Institute of Technology, Stockholm, 10044, Sweden
Email: {jinfeng, medard}@mit.edu

Abstract—Data transmission over multiple-hop networks is impaired by random deleterious events, and characterizing the probability of error for the end-to-end transmission is challenging as the size of networks grows. Adams et al. showed that, when re-encoding at intermediate nodes is enabled, coded transmission over tandem/parallel links can be reduced to a single equivalent link with a specified probability function. Although iterative application of the tandem/parallel reduction techniques in alternation can simplify the task, they are generally not sufficient to reduce an arbitrary network to a single link. In this paper, we propose upper- and lower-bounding processes to bound the end-to-end probability distribution of a network by combining the parallel/tandem link reduction with the structure of flows over the network. We evaluate the performance of the proposed bounding methods at the 99% success rate of end-to-end data transmission over randomly generated acyclic networks. The numerical results demonstrate that our bounding approaches enable us to characterize a network by a single probability function to a very good precision.

Index Terms—network reduction, multi-hop networks, network coding, coded transmission

I. INTRODUCTION

Data transmission over large networks is impaired by random deleterious events, such as packet losses caused by congestion and decoding errors owing to noise or collision. Reliable end-to-end transmission relies largely on various acknowledgement and retransmission schemes at the cost of transmission efficiency, since packet-wise (or block-wise) acknowledgement is needed on a end-to-end or even link-by-link basis. Numerous research efforts have been devoted to characterizing the fundamental limits of data transmission over lossy networks and to improving its efficiency from different aspects. We only list a very sparse sampling here. Dana et al. in [1] consider the use of linear network coding over packets and construct a network model based on correlated erasure links. Assuming the destination node has side information on each packet loss event and allowing packets broadcast from one node to its neighbors, the authors show that linear network codes achieve the capacity of such networks. Lun et al. in [2] propose a framework to translate a lossy unicast or multicast network into a lossless packet network by applying random linear network coding (RLNC) [3] and performing RLNC re-encoding at intermediate nodes. Assuming independent Poisson packet arrivals at each node and the number of packets is large, the probability of RLNC decoding error is characterized by the delay, rate, and the network capacity. Unlike [1], no side information is required to achieve capacity. While [2] does not consider the practical constraint of buffer size at intermediate nodes, Haeupler and Médard show in [4] that RLNC is asymptotically capacity-achieving even if intermediate nodes may only store one coded packet. Xiao et al. in [5] investigate the delay in packet erasure networks where RLNC is used in a rateless fashion, and its performance is upper bounded by a single packet erasure link generated based on all of the links in the minimum cut.

In general, if we choose a block of bits as the basic data unit for transmission, the behavior of each link in a network is simply characterized by a probability function that associates the probability of error with the number of data units transmitted within unit time. This data unit is most analogous to a packet, although the size of packets may vary depending on the link quality: a packet may carry several data units, and a data unit may be split into several packets. Hereafter we simply use packet when we actually refer to the data unit.

Considering the network abstraction where the maximum data rate and packet loss probability across each link are limited by local constraints, and assuming that packet losses are independent, it is interesting to ask whether the behavior of data transmission over an entire network or a collection of links can be fully described by a single probability function. In [6] Adams et al. have proven that links which are connected in tandem or in parallel can be reduced to a single equivalent link with a specified probability function. These two reduction operations serve to make networks simpler to model and study, but are not sufficient to reduce an arbitrary network. To tackle this difficulty, we propose upper- and lower-bounding methods by combining the parallel/tandem link reduction with the structure of data flows over the underlying networks. Numerical results over randomly generated networks demonstrate the effectiveness of the proposed methods.

The rest of this paper is organized as follows. In Sec. II we describe the system model and the parallel/tandem link reduction methods. We focus on flows over cuts in Sec. III to construct cut-based upper and lower bounds. Our node-reduction based upper bounds are presented in Sec. IV, and routing-based lower bounds are in Sec. V. We present the numerical results in Sec. VI and conclude in Sec. VII.

* N. Sweeting was a student with Hunter College High School, New York.

This paper has been accepted for publication in Proc. IEEE ICC 2015, London, UK, Jun. 2015. Copyright will be transferred to IEEE without notice.
II. System Model

We represent a network by its underlying directed graph \( \mathcal{N} = \{ \mathcal{V}, \mathcal{E} \} \) where \( \mathcal{V} \) is the set of all vertices and \( \mathcal{E} \) is the collection of edges. For some \( u, v \in \mathcal{V} \), data transmission from \( u \) to \( v \) is possible if and only if \((u, v) \in \mathcal{E}\). If there are multiple edges among two nodes, we apply subscripts to distinguish them. For each \( v \in \mathcal{V} \), we denote the \textbf{parents set} of \( v \) by
\[
\mathcal{I}(v) \triangleq \{ u \in \mathcal{V} | (u, v) \in \mathcal{E} \},
\]
the \textbf{children set} of \( v \) by
\[
\mathcal{O}(v) \triangleq \{ u \in \mathcal{V} | (v, u) \in \mathcal{E} \},
\]
and the set of edges incident to \( v \) by
\[
\mathcal{E}(v) \triangleq \{ e \in \mathcal{E} | e = (u, v), \text{ or } e = (v, u) \text{ for some } u \in \mathcal{V} \}. \]

In this paper we only focus on a single unicast transmission over acyclic networks. Extension to more general setups are left to future work. Under this scenario, the networks we are investigating in this paper have the following properties.

- For the \textbf{source} node \( v_s \) and the \textbf{destination} node \( v_d \), we have \( |\mathcal{I}(v_s)| = 0, |\mathcal{O}(v_s)| > 0 \) and \( |\mathcal{O}(v_d)| = 0, |\mathcal{I}(v_d)| > 0 \).
- Every node is reachable from \( v_s \) and reverse-reachable from \( v_d \) (otherwise the node and its connected edges can be removed from the graph).
- For link \( e_i \in \mathcal{E}, n_i \) denotes the maximum number of packets that can be transmitted over the link within the time constraint, and \( \xi_i \) is the probability that a packet is dropped independently at random.

Therefore data transmission over \( e_i \in \mathcal{E} \) is characterized either by the Probability Mass Function (PMF) \( \lambda_1 \in [0, 1]^{n_i+1} \) or by the Complementary Cumulative Distribution Function (CCDF) \( \lambda_i \in [0, 1]^{n_i+1} \), where \( \lambda_i(k) \) describes the probability that link \( e_i \) can successfully transmit exactly \( k \) packets per delay constraint, and \( \Lambda_i(k) \) describes the probability that at least \( k \) packets are successfully transmitted. Given \((n_i, \xi_i)\), the PMF \( \lambda_i \) is defined by a binomial distribution, i.e.,
\[
\lambda_i(k) = \binom{n_i}{k} \xi_i^{n_i-k}(1-\xi_i)^k, \quad \text{for } 0 \leq k \leq n_i, \quad 0, \quad \text{otherwise.} \tag{1}
\]
We can easily convert PMFs to/from CCDFs via the following one-to-one mapping
\[
\Lambda_i(k) = \sum_{j=k}^{n_i} \lambda_i(j), \quad \Lambda_i(k) = \Lambda_i(k) - \Lambda_i(k+1), \tag{2}
\]
where \( \Lambda_i(k) = 0, \forall k > n_i \) by default.

The tandem/parallel reduction techniques developed in [6] state that we can describe a two-link tandem/parallel network, as shown in Fig. 1, by a single PDF/CCDF. Define \( S, S_1, \) and \( S_2 \) the number of packets successfully transmitted across the network, and across the two individual links, respectively. The tandem network can be described by a single CCDF
\[
\Lambda(k) \triangleq P(S \geq k) = P(\min\{S_1, S_2\} \geq k) = P(S_1 \geq k)P(S_2 \geq k) = \Lambda_1(k)\Lambda_2(k),
\]

or equivalently, \( \Lambda = \Lambda_1 \odot \Lambda_2 \) where \( \odot \) denotes element-wise multiplication (Hadamard product) after zero-padding the shorter of the two CCDFs. Similarly, the parallel network with PMFs \( \lambda_1 \) and \( \lambda_2 \) can be described by a single PMF
\[
\lambda(k) \triangleq P(S = k) = P(S_1 + S_2 = k) = \sum_{j=0}^{k} P(S_1 = j)P(S_2 = k-j) = \sum_{j=0}^{k} \lambda_1(j)\lambda_2(k-j),
\]

or equivalently \( \lambda = \lambda_1 \ast \lambda_2 \) where \( \ast \) denotes convolution.

III. Flow-Cut Bounds

The two-link tandem/parallel reduction techniques can be straightforwardly extended to multiple links by induction. We can iteratively apply the tandem- and parallel-link reduction techniques in alternation to simplify the calculation of end-to-end PMFs/CCDFs. We should note, however, that the tandem/parallel reduction operations will not simplify arbitrary networks to a single distribution. Fig. 2 depicts a simple network that cannot be simplified in this way; no two links form a purely parallel or tandem structure, because of link \( e_3 \). Indeed, this network is the smallest network (in terms of the number of nodes/edges) that can’t be fully reduced. As the number of nodes and edges grows, the possibility that the tandem/parallel reduction operations are sufficient for network reduction will decrease. We therefore need new approaches to tackle general network topologies that can’t be fully reduced.

Definition 1 (Upper and Lower Bounds of a CCDF): Let \( S \) be the number of successfully delivered packets and \( \Lambda \) be its CCDF, if for all non-negative integers \( k < E[S] \), we have
\[
\Lambda_L(k) \leq \Lambda(k) \leq \Lambda_U(k), \tag{3}
\]
then \( \Lambda_L \) is a lower bound and \( \Lambda_U \) is an upper bound.

This definition only focus on \( k < E[S] \), motivated by the fact we are only interested in operation regimes where the rate of successful is large (say larger than 50%).

Definition 2 (Flow across a Cut): Let \( C \) be a cut of the directed acyclic graph and \( \mathcal{E}(C) \) be the set of edges that cross \( C \) from the source side to the destination side. The flow across

![Fig. 1. Two links in tandem (left) and in parallel (right).](image1)

![Fig. 2. The smallest network where the tandem/parallel link reduction fails.](image2)
the cut $\mathcal{C}$, defined as the number of successfully delivered packets over the cut $\mathcal{C}$ within unit time, is therefore

$$S_\mathcal{C} = \sum_{i \in \mathcal{E} (\mathcal{C})} S_i.$$  

Assuming $\mathcal{E}(\mathcal{C}) = \{1, 2, \ldots, m\}$, its associated PMF $\lambda_\mathcal{C}$ is

$$\lambda_\mathcal{C} = \lambda_1 * \lambda_2 * \cdots * \lambda_m,$$

where the equality is due to the parallel-link reduction by regarding links in $\mathcal{E}(\mathcal{C})$ as a parallel network.

**Proposition 1:** Let $\mathcal{C}_d$, $d = 1, 2, \ldots, c$, be all the cuts separating the source node $v_s$ and the destination node $v_d$, each associated with a flow $S_{\mathcal{C}_d}$ and a CCDF $\Lambda_{\mathcal{C}_d}$, and define

$$\Lambda_{\text{all-cuts}} \triangleq \Lambda_{\mathcal{C}_1} \circ \Lambda_{\mathcal{C}_2} \circ \cdots \circ \Lambda_{\mathcal{C}_c},$$

$\Lambda_{v_s \rightarrow v_d}$, which describes the end-to-end data transmission, is lower bounded by $\Lambda_{\text{all-cuts}}$ and upper bounded by $\Lambda_{\mathcal{C}_d} \forall d$.

**Proof:** Denoting $D$ the number of successfully received data units at the destination node $v_d$, we have

$$D = \min\{S_{\mathcal{C}_1}, S_{\mathcal{C}_2}, \ldots, S_{\mathcal{C}_c}\},$$

where the equality comes from the fact that information passing through the network goes through every cut in order to reach the destination. Therefore any cut $\mathcal{C}_d \forall d$ provides a valid upper bound $\Lambda_{\mathcal{C}_d}$. To prove the lower bound, we need to show that for all non-negative integers $k < E[D]$ where $E[D]$ is the mean associated with $\Lambda_D$, we have

$$\Lambda_{\text{all-cuts}}(k) \leq \Lambda_D(k).$$

Intuitively, $\Lambda_{\text{all-cuts}}$ neglects the dependence between all cuts and therefore represents a tandem network by connecting all cuts in serial. The formal proof of (5) is in Appendix A.

We refer to $\Lambda_{\text{all-cuts}}$ as the All-Cuts lower bound and $\Lambda_{\text{min}}$ as the Min-Cut upper bound, where $\Lambda_{\text{min}}$ is the cut whose average throughput is smallest among all cuts.

**Remark 1:** It is interesting to compare our Min-Cut upper bound to the one proposed in [5, Proposition 1], where all the erasure links $(n_i, \xi_i)$ crossed by the minimum cut are modeled by a single erasure channel $(n, \xi)$, where $n = \sum_i n_i$ and $\xi = \sum_i \frac{n_i}{n} \xi_i$. From Proposition 2 we can see that [5, Proposition 1] is accurate up to the first moment (the mean) $n(1 - \xi)$ but provides a larger variance $n\xi(1 - \xi)$.

**Proposition 2:** Given $s$ independent binomial random variables $X_i \sim B(n_i, 1 - \xi_i)$, $i = 1, \ldots, s$, and denoting

$$X = \sum_{i=1}^{s} X_i, \quad n = \sum_{i=1}^{s} n_i, \quad \xi = \sum_{i=1}^{s} \frac{n_i}{n} \xi_i,$$

we have

$$E(X) = n(1 - \xi), \quad \text{Var}(X) \leq n\xi(1 - \xi),$$

where the equality holds if and only if $\xi_1 = \ldots = \xi_s$.

**Proof:** Since $X_i$ are independent, we have

$$E(X) = \sum_{i=1}^{s} E(X_i) = \sum_{i=1}^{s} n_i(1 - \xi_i) = n(1 - \xi),$$

$$\text{Var}(X) = \sum_{i=1}^{s} \text{Var}(X_i) = \sum_{i=1}^{s} n_i \xi_i(1 - \xi_i).$$

W.l.o.g., assuming $0 \leq \xi_1 \leq \ldots \leq \xi_s \leq 1$, we have $1 \geq 1 - \xi_1 \geq \cdots \geq 1 - \xi_s \geq 0$. By the Chebyshev Sum Inequality,

$$\frac{1}{n} \sum_{i=1}^{s} n_i \xi_i(1 - \xi_i) \leq \left( \frac{1}{n} \sum_{i=1}^{s} n_i \xi_i \right) \left( \frac{1}{n} \sum_{j=1}^{s} n_j (1 - \xi_j) \right),$$

$$= \xi \left( 1 - \sum_{j=1}^{s} \frac{n_j \xi_j}{n} \right) = \xi(1 - \xi),$$

where the equality holds if and only if $\xi_1 = \ldots = \xi_s$. Substituting (8) into (9) and multiplying both sides by $n$, we get (6).

**IV. NETWORK REDUCTION UPPER BOUNDS**

We can also generate upper bounds by first altering the network structure while preserving its minimum cut, and then applying parallel/tandem link reduction. Firstly, we define the node reduction operations more precisely.

**Definition 3:** The $n$-node reduction function $f(N, v)$, where $v$ has $\mathcal{I}(v) = \{v_in\}$ and $\mathcal{O}(v) = \{o_1, o_2, \ldots, o_n\}$, maps $\mathcal{N} \{V, E\}$ to $\mathcal{N'} \{V', E'\}$, where

$$\mathcal{V}' = \mathcal{V} \setminus \{v\}$$

and $\mathcal{E}' = (\mathcal{E} \setminus \mathcal{E}(v)) \bigcup \sum_{i=1}^{n} (v_{in}, o_i)$.

We associate each new link $(v_{in}, o_i)$ with a CCDF

$$\Lambda_{v_{in} \rightarrow o_i} = \Lambda_{v_{in} \rightarrow v} \circ \Lambda_{v \rightarrow v_{out}}.$$

**Definition 4:** The $n$-node reduction function $g(N, v)$, where $v$ has $\mathcal{O}(v) = \{v_{out}\}$ and $\mathcal{I}(v) = \{i_1, i_2, \ldots, i_n\}$, maps $\mathcal{N} \{V, E\}$ to $\mathcal{N'} \{V', E'\}$, where

$$\mathcal{V}' = \mathcal{V} \setminus \{v\}$$

and $\mathcal{E}' = (\mathcal{E} \setminus \mathcal{E}(v)) \bigcup \sum_{k=1}^{n} (i_k, v_{out})$.

We associate each new link $(i_k, v_{out})$ with a CCDF

$$\Lambda_{i_k \rightarrow v_{out}} = \Lambda_{i_k \rightarrow v} \circ \Lambda_{v \rightarrow v_{out}}.$$

A node $v$ is said to be $1$-$n$-reducible if it has $|\mathcal{I}(v)| = 1$, or $n$-reducible if it has $|\mathcal{O}(v)| = 1$. The $1/n/1$-node reduction operations can be visualized as in Fig. 3, where a $1/n$- ($n$-1)-reducible node is firstly identified, then copied with its incoming (outgoing) edge, and finally removed by applying tandem-link reduction.

**Proposition 3:** For network $\mathcal{N}$, $f(N, v)$ and $g(N, v)$ provide upper bounds.

**Proof:** It suffices to notice that for any transmission task supported by $\mathcal{N}$, we can find a corresponding transmission protocol on the reduced network with the same or relaxed constraints, due to the underlying node/edge duplication operation by $f(N, v)$ and $g(N, v)$.

**Proposition 4:** Any unicast acyclic network $\mathcal{N}$ has at least one node $v$ such that $v$ is $1/n$-reducible and at least one node $u$ such that $u$ is $n$-1-reducible.

**Proof:** See the appendix B.

We can fully reduce a network to a single link by repeatedly applying $1/n$-1-node reduction and tandem/parallel link reductions, and hence provide upper bounds for the
original network. Disadvantages are the potentially relaxed transmission constraints at the reduced nodes and that it does not tell us in which order these 1n/n1-reducible nodes should be reduced. We propose two strategies to select the node to be first reduced: the 1n/n1-reducible node with the highest incoming/outgoing link capacity (termed NR-Abs bound) or the one with highest ratio between incoming and outgoing link capacity (termed NR-Ratio bound).

**Definition 5:** The **capacity ratio** of a 1n/n1-reducible node $v$ is defined based on the average throughput across its incoming/outgoing edges, i.e.,

$$\overline{C}_v = \frac{C_{(v,v_0)}}{\sum_{i=1}^{n} C_{(v,v_i)}}$$

or $\overline{C}_v = \frac{\sum_{i=1}^{n} C_{(v_0,v_i)}}{C_{(v,v_0)}}$.

**Proposition 5:** If $\overline{C}_v \geq 1$ for some 1n/n1-reducible node $v$, the minimum cut of the NR-Ratio reduced network is the same as that of the original network.

Proof omitted here since it is intuitive to see that the minimum cut is always preserved in NR-Ratio when $\overline{C}_v \geq 1$.

We can create a hybrid approach with some predetermined threshold $t$: applying NR-Ratio to reduce all nodes with $\overline{C}_v > t$ and then apply NR-Abs reduction for the remaining 1n/n1-reducible nodes. Choosing $t = 0$ will be identical to NR-Ratio and choosing $t = \infty$ will be identical to NR-Abs.

**V. FORD-FULKERSON BASED LOWER BOUNDS**

The Ford-Fulkerson algorithm [7] computes the optimal routing paths in a flow network. To adapt the algorithm for our purposes, we use the average throughput of each link as the flow value, apply the Ford-Fulkerson algorithm to find all the feasible flows, and then split the network into disjoint paths using conservation of flow as shown in Fig. 4. Each path forms a tandem network, and all paths form a parallel network. Then we can use the basic reduction operations to reduce the network into a single link. Given a path with flow $y_j$ in its tandem network, a link that originated from edge $e_i$ inherits its erasure probability $\xi_i$ and a share of its rate $n'_{i,j} = \frac{y_j}{\sum_i y_i}$ to ensure integrality constraint. This approach is termed FF-Flow, as shown in Fig. 4 (middle). We can also split a link according to the shares of each flow that pass through it, i.e., $n'_{i,j} = \frac{y_j}{\sum_k y_k n_i}$ where $\sum_k y_k$ is the total flow that passes through the link $e_i$. This approach is named FF-Split and illustrated in Fig. 4 (right).

Intuitively, the Ford-Fulkerson algorithm-based reduction yields lower bounds because the algorithm provides us with a network protocol on the original network that can produce the corresponding probability function.

**VI. NUMERICAL ILLUSTRATION**

To understand the general performance of all our proposed upper and lower bounds, we evaluate their end-to-end probability function over the 4-node irreducible network and over randomly generated networks. For a predefined network size (in nodes), the probability that there is a directed edge from one node to the other is set to 1/2 and the rate of each link is uniformly chosen from $[100, 1000]$ with erasure probability randomly chosen from within a predefined range. Once the network has been populated, networks with cycles or isolated nodes will be discarded. Furthermore, we only focus on cases where tandem/parallel reduction are not sufficient as we would not need these bounding methods otherwise.

**A. Error Probability over a Four-Node Test Network**

We simulate the end-to-end error probability via $10^7$ Monte-Carlo trials over the smallest irreducible network where all links have the same erasure probability $\xi=0.1$ but different rates, indicated by the numbers along the corresponding edges as in Fig. 5. In this test network, the NR-Abs bound and the NR-Ratio bound generated as in Fig. 3 are tighter than the MinCut upper bound. The FF-Flow lower bound is identical to FF-Split and they are better than the All-Cuts in some regions.

**B. Random Networks: Gap from the Best Upper/Lower Bound**

To evaluate the tightness of our bounds in general network settings, we compare their performance over randomly generated acyclic networks by the highest end-to-end data rate they can support with no less than 99% success probability. We call the corresponding rate $K_{99}$. In Fig. 6 we evaluate the $K_{99}$ of all the lower bounds against the best upper bound (i.e., the smallest $K_{99}$ produced by all the upper bounds) and...
plot the mean value and the corresponding range based on 1000 trials for each network size and erasure probability range. The All-Cuts lower bound is always within 1% of the best upper bound over random networks with different sizes and erasure probability. The FF-Split lower bound improves FF-Flow uniformly. FF-Split (FF-Flow) provides a gap of less than 1% (2%) on average and less than 3% (6%) in the worst case\(^1\), at least for the settings as we have demonstrated. Their performances degrade slightly with increasing network size and the erasure probability.

The Min-Cut upper bound provides a gap of less than one percent, which is much better than the NR-Abs and the NR-Ratio bounds over random networks as shown in Fig. 7, where the erasure probability is chosen from the range [3%, 10%]. Although NR-Abs and NR-Ratio may produce better bounds than the Min-Cut, they may also relax the network constraints, since even a highly varied network often does not provide any nodes with a capacity ratio larger than 1. Therefore, they may produce a loose bound, as indicated by the excessive range shown in Fig. 7 (above). Their mean and variance increase as the size of networks grows. Performance for other erasure probability ranges are similar and therefore omitted here.

C. Random Networks: Chances as the Best Bound

In Fig. 8 we count the instances in which each bounding method produces the best bound, as measured by \(K_{99}\). With very high probability All-Cuts provides the best lower bound (> 95%) and Min-Cut provides the best upper bound (> 99%). The FF-Split produces the best lower bound with about 20% probability when network size is small. The NR-Abs and
the NR-Ratio upper bounds are the best upper bound with high probability when the network constraints are preserved, although the probability to produce a loose bound is also large, as indicated by the large range in Fig. 7.

VII. CONCLUSIONS

In this work, we propose several lower and upper bounds to characterize the end-to-end transmission probability function. Our best lower bound yields a gap smaller than one percent in throughput from the best upper bound over randomly generated acyclic networks. This justifies our efforts by describing the end-to-end data transmission over lossy networks with a single probability function to high precision.

There are several ways to improve our proposed upper and lower bounds. For example, one can combine the Min-Cut and the NR-Ratio upper bounds to construct a new upper bound: we first reduce the network as in NR-Ratio until there is no node with input-output capacity ratio higher than 1, and then apply Min-Cut to the reduced network. We may also combine the 1/n1-node reduction and the All-Cuts lower bound to provide a good approximation that always falls between the Min-Cut upper bound and the All-Cuts lower bound. Other approximations can be found in [8].

APPENDIX A

PROOF OF THE ALL-CUTS LOWER BOUND

If no edge appears in more than one cut, i.e., all \( C_d \), \( \forall d \) are independent, all the cuts form a tandem network. By tandem-link reduction we have

\[
\Lambda_D = \Lambda_{C_1} \odot \cdots \odot \Lambda_{C_n} = \Lambda_{\text{all-cuts}}.
\]

Assuming \( C_1 \) and \( C_2 \) share some common links, we introduce three independent random variables \( Z_0, Z_1, Z_2 \) such that

\[
S_{C_1} = Z_0 + Z_1, \quad S_{C_2} = Z_0 + Z_2,
\]

where \( Z_0 \) represents the flow over the common links and \( Z_1 \) and \( Z_2 \) represent flows over the rest links in \( C_1 \) and \( C_2 \), respectively. We have

\[
\Lambda_{C_1}(k) = P(Z_0 + Z_1 \geq k) = \sum_i P(Z_1 \geq k - i | Z_0 = i) P(Z_0 = i) = \sum_i P(Z_1 \geq k - i) P(Z_0 = i) = \sum_i P(Z_0 = i) \Lambda_{Z_1}(k - i),
\]

where the second last equality comes from the fact that \( Z_0 \) and \( Z_1 \) are independent. Similarly we can show that

\[
\Lambda_{C_2}(k) = \sum_i P(Z_2 \geq k - i) P(Z_0 = i) = \sum_i P(Z_0 = i) \Lambda_{Z_2}(k - i).
\]

On the other hand, denoting \( S_m = \min\{S_{C_1}, S_{C_2}\} \), we have

\[
\Lambda_{S_m}(k) = P(S_m \geq k) = P(\min\{Z_1, Z_2\} + Z_0 \geq k) = \sum_i P(Z_1 \geq k - i, Z_2 \geq k - i | Z_0 = i) P(Z_0 = i) = \sum_i P(Z_0 = i) \Lambda_{Z_1}(k - i) \Lambda_{Z_2}(k - i).
\]

Since \( \Lambda_{Z_1}(j) \) and \( \Lambda_{Z_2}(j) \) are monotonically decreasing, we can show by following the Chebyshev Sum Inequality that for all feasible \( k \) (as long as \( \Lambda(k) > 0 \)),

\[
\Lambda_{C_1}(k) \Lambda_{C_2}(k) \leq \Lambda_{S_m}(k).
\]

By grouping cuts that share common links and rearranging \( D \) in such a way that

\[
D = \min\{\cdots \min\{S_{C_1}, S_{C_2}\}, S_{C_3}\} \cdots S_{C_n}\},
\]

we can apply the above results iteratively and prove (5).

APPENDIX B

PROOF OF PROPOSITION 4

Let us consider the nodes \( v \in O(v_0) \). If there is only one such node (call it \( v_0 \)), it must be \( 1n \)-reducible, since otherwise it would have an input which is not the source. Say it is connected to node \( u \). If we trace the input of node \( u \) by traveling in reverse along the edges of the network, we must eventually reach the source, which means we must pass through \( v \). Thus \( u \), which has \( v \in O(u) \), is part of some path originating at \( v \). This means we have found a cycle; a contradiction. Thus we need only consider the case in which there are multiple nodes \( v \in O(v_0) \). Using the same reasoning as above, for each such \( v \) we can trace a non-source input back to some other \( v' \in O(v_0) \). However, if we do the same for \( v' \), we find that any input not directly from the source must originate from some \( v'' \in O(v_0) \). If we continue this process, we must at some point reach a repeated node, since the network is finite. This means we have found a cycle, a contradiction. So there must be some \( v \in O(v_0) \) such that \( \mathcal{I}(v) = v \). We can make the completely analogous “dual” argument using the sink to prove the existence of an \( n1 \)-reducible node.

ACKNOWLEDGMENT

This work was funded in part by the Swedish Research Council (VR), the VT iDirect, and the MIT Wireless Center. This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under award No. FA9550-13-1-0023.

REFERENCES