

On Equivalence Between Network Topologies

Tracey Ho

Department of Electrical Engineering
California Institute of Technology
tho@caltech.edu;

Michelle Effros

Departments of Electrical Engineering
California Institute of Technology
effros@caltech.edu;

Shirin Jalali

Center for Mathematics of Information
California Institute of Technology
shirin@caltech.edu;

Abstract—One major open problem in network coding is to characterize the capacity region of a general multi-source multi-demand network. There are some existing computational tools for bounding the capacity of general networks, but their computational complexity grows very quickly with the size of the network. This motivates us to propose a new hierarchical approach which finds upper and lower bounding networks of smaller size for a given network. This approach sequentially replaces components of the network with simpler structures, i.e., with fewer links or nodes, so that the resulting network is more amenable to computational analysis and its capacity provides an upper or lower bound on the capacity of the original network. The accuracy of the resulting bounds can be bounded as a function of the link capacities. Surprisingly, we are able to simplify some families of network structures without any loss in accuracy.

I. INTRODUCTION

Finding the network coding capacity of networks with general topologies and communication demands is a challenging open problem, even for networks consisting of noiseless point-to-point links. Information theoretic inequalities can be used to bound network capacities, but it is in general a complex task to find the best combination of inequalities to apply. While various bounds (e.g. [1], [2], [2], [3]) can be obtained by clever choices of inequalities, we would like to have systematic general techniques for bounding the capacity in arbitrary network problems. We hope to derive these bounds in a way that allows us to bound the accuracy of the obtained bounds and to trade off tightness and computation. The LP outer bound [4], which gives the tightest outer bound implied by Shannon-type inequalities and has been implemented as the software programs Information Theoretic Inequalities Prover (ITIP) [5] and XITIP [6], has complexity exponential in the number of links in the network and can thus only be used to compute capacity bounds for relatively small problem instances. Inner bounds can be obtained by restricting attention to scalar linear, vector linear or abelian codes e.g. [7], [8], but the complexity of such approaches also grows quickly in the network size. This motivates us to seek systematic tools for bounding the capacity of a network by the capacity of another network with fewer links and characterizing the difference in capacity.

In this paper we introduce a novel approach for analyzing capacity regions of acyclic networks consisting of capacitated noiseless links with general demands. Inspired by [9], we employ a hierarchical network analysis that replaces pieces

of the network by equivalent or bounding models with fewer links. At each step of the process, one component of the network is replaced by a simpler structure with the same inputs and outputs. The model is an upper bounding model if all functions that can be implemented across the original network can also be implemented across the model. The model is a lower bounding model if all functions that can be implemented across that model can also be implemented across the given component. If the same model is both an upper bounding model and a lower bounding model for a given component, then the component and its model are equivalent. Where possible, we try to find upper and lower bounds that have identical structures, since bounding the accuracy of the resulting capacity bounds is easier when the topologies of the upper and lower bounding networks match.

The organization of this paper is as follows. Section II describes the system model. The problem of finding equivalent or bounding networks of smaller size and the properties of such networks is discussed in Section III. Sections IV and V describe a variety of operations for finding such networks. Section VI treats accuracy bounds. The networks considered in this paper are assumed to be acyclic. The effect of cycles and delay is discussed in Section VII. Finally, Section VIII concludes the paper.

II. SYSTEM MODEL

We mainly use the model and notation introduced in [10]. The network is modeled by an acyclic directed graph $\mathcal{N} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the set of nodes and links, respectively. Each directed link $e = (v_1, v_2) \in \mathcal{E}$ represents a noiseless link of capacity C_e between the nodes v_1 and v_2 in \mathcal{N} . For each node v , let $\text{In}(v)$ and $\text{Out}(v)$ denote the set of incoming and outgoing links of node v respectively.

We assume that the source nodes (\mathcal{S}) and sink nodes (\mathcal{T}) are distinct, i.e., $\mathcal{S} \cap \mathcal{T} = \emptyset$, and each source (sink) node has only outgoing (incoming) links. There is no loss of generality in this assumption since any network that violates these conditions can be modified to form a network that satisfies these conditions and has the same capacity. After these modifications, each node $v \in \mathcal{V}$ falls into one of the following categories: i) source nodes (\mathcal{S}), ii) sink nodes (\mathcal{T}) and iii) relay nodes (\mathcal{I}). Relay nodes have both incoming and outgoing links, and they do not observe any external information. Their role is to facilitate data transfer from the source nodes to the sink nodes.

A subnetwork $\mathcal{N}_s = (\mathcal{V}_s, \mathcal{E}_s)$ of network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ is constructed based on a subset of relay nodes $I_s \subset I$ as follows. The set of sources and sinks of the subnetwork \mathcal{N}_s are defined as $\mathcal{S}_s = \{v \in \mathcal{V} : v \notin I_s, \exists v' \in I_s, (v, v') \in \mathcal{E}\}$, and $\mathcal{T}_s = \{v \in \mathcal{V} : v \notin I_s, \exists v' \in I_s, (v', v) \in \mathcal{E}\}$. Then, $\mathcal{V}_s = \mathcal{S}_s \cup I_s \cup \mathcal{T}_s$, and $\mathcal{E}_s = \{e \in \mathcal{E} : e = (v, v'), v, v' \in \mathcal{V}_s\}$.

A coding scheme of block length n for this network is described as follows. Each source node $s \in \mathcal{S}$ observes some message $M_s \in \mathcal{X}_s = \{1, 2, \dots, 2^{nR_s}\}$. Each sink node $t \in \mathcal{T}$ is interested in recovering some of the messages that are observed by source nodes. Let $\beta(t) \subseteq \mathcal{S}$ denote the set of source nodes that the node t is interested in recovering. The coding operations performed by each node can be categorized as follows

- 1) Encoding functions performed by the source nodes: For each $s \in \mathcal{S}$, and $e \in \text{Out}(s)$, the encoding function corresponding to link e is described as

$$g_e : \mathcal{X}_s \rightarrow \{1, 2, \dots, 2^{nC_e}\}. \quad (1)$$

- 2) Relay functions performed at relay nodes: If $v \notin \mathcal{S} \cup \mathcal{T}$, then for each $e \in \text{Out}(v)$, the relay function corresponding to the link e is described as

$$g_e : \prod_{e' \in \text{In}(v)} \{1, 2, \dots, 2^{nC_{e'}}\} \rightarrow \{1, 2, \dots, 2^{nC_e}\}. \quad (2)$$

- 3) Finally, for each $t \in \mathcal{T}$, and each $s \in \beta(t)$, a decoding function is defined as

$$g_t^s : \prod_{e \in \text{In}(t)} \{1, 2, \dots, 2^{nC_e}\} \rightarrow \mathcal{X}_s. \quad (3)$$

A rate vector \mathbf{R} corresponding to the set $\{R_s\}_{s \in \mathcal{S}}$ is said to be achievable on network \mathcal{N} , if for any $\varepsilon > 0$, there exists n large enough and a coding scheme of block length n such that for all $t \in \mathcal{T}$ and $s \in \beta(t)$

$$\mathbb{P}(\hat{M}_s^{(t)} \neq X_s) \leq \varepsilon, \quad (4)$$

where $\hat{M}_s^{(t)}$ denotes the reconstruction of message M_s at node t . Let $\mathcal{R}(\mathcal{N})$ denote the set of achievable rates on network \mathcal{N} .

Throughout the paper, vectors are denoted by bold upper-case letters, e.g. \mathbf{A} , \mathbf{B} , etc. Sets are denoted by calligraphic upper-case letters, e.g. \mathcal{A} , \mathcal{B} , etc. For a vector $\mathbf{A} = (a_1, a_2, \dots, a_n)$ of length n and a set $\mathcal{F} \subset \{1, 2, \dots, n\}$, $\mathbf{A}_{\mathcal{F}}$ denotes a vector of length $|\mathcal{F}|$ formed by the elements of the vector \mathbf{A} whose indices are in the set \mathcal{F} in the order they show up in \mathbf{A} .

III. EQUIVALENT AND BOUNDING NETWORKS

The problem we consider is defined formally as follows. For a given network \mathcal{N} , we wish to find a network \mathcal{N}' with fewer links for which the set of achievable rates either bounds $\mathcal{R}(\mathcal{N})$ from below ($\mathcal{R}(\mathcal{N}') \subseteq \mathcal{R}(\mathcal{N})$), bounds $\mathcal{R}(\mathcal{N})$ from above ($\mathcal{R}(\mathcal{N}) \subseteq \mathcal{R}(\mathcal{N}')$) or describes it perfectly ($\mathcal{R}(\mathcal{N}) = \mathcal{R}(\mathcal{N}')$). We take a hierarchical approach, sequentially applying operations to simplify the given network. Following [9], each operation replaces a subnetwork of the network with a bounding model. Subnetwork \mathcal{N}_2 is an upper bounding model for a

subnetwork \mathcal{N}_1 with the same number of input (source) and output (sink) nodes (written $\mathcal{N}_1 \subseteq \mathcal{N}_2$) if all functions $\{f_t\}_{t \in \mathcal{T}}$ of sources that can be reconstructed at the sinks of \mathcal{N}_1 can also be reconstructed at the sinks of \mathcal{N}_2 . Here, each function f_t , for $t \in \mathcal{T}$, is a function of the information sources that are available at source nodes. Subnetworks \mathcal{N}_1 and \mathcal{N}_2 are equivalent if $\mathcal{N}_1 \subseteq \mathcal{N}_2$ and $\mathcal{N}_2 \subseteq \mathcal{N}_1$.

When deriving upper and lower bounding networks, it is desirable to find upper and lower bounding networks that have the same topologies. In this case, we can bound the difference between the capacity of a network \mathcal{N} and capacities $\mathcal{R}(\mathcal{N}_l)$ and $\mathcal{R}(\mathcal{N}_u)$ of lower and upper bounding networks \mathcal{N}_l and \mathcal{N}_u using a bound from [11]. Note that by having identical graphs, we also require that all links have non-zero capacity in both networks.

For comparing two networks \mathcal{N}_l and \mathcal{N}_u which have identical topologies, define the difference factor between \mathcal{N}_l and \mathcal{N}_u as

$$\Delta(\mathcal{N}_l, \mathcal{N}_u) \triangleq \max_{e \in \mathcal{E}} \frac{C_e(\mathcal{N}_u)}{C_e(\mathcal{N}_l)}, \quad (5)$$

where $C_e(\mathcal{N}_l)$ and $C_e(\mathcal{N}_u)$ denote the capacities of the link e is \mathcal{N}_l and \mathcal{N}_u respectively. Note that $\Delta(\mathcal{N}_l, \mathcal{N}_u) \geq 1$. Let $\mathcal{R}_l(\mathcal{R}_u)$ denote the capacity region of \mathcal{N}_l (\mathcal{N}_u). Then

$$\mathcal{R}_l \subseteq \mathcal{R} \subseteq \mathcal{R}_u, \quad (6)$$

while

$$\mathcal{R}_u \subseteq \Delta(\mathcal{N}_l, \mathcal{N}_u) \mathcal{R}_l. \quad (7)$$

IV. BASIC SIMPLIFICATION OPERATIONS

One of the simplest operations for deriving an upper-bounding network for a given network is merging nodes. Coalescing two nodes is equivalent to adding two links of infinite capacity from each of them to the other one. This is precisely the approach employed in cut-set bounds. Because of these infinite-capacity links, combining nodes, unless done wisely, potentially can result in very loose bounds. However, we show that in some cases, nodes can be combined without affecting the network capacity. One simple example is when the sum of the capacities of the incoming links of a node v is less than the capacity of each of its outgoing links. In this situation, the node can be combined with all nodes w such that $(v, w) \in \mathcal{E}$.

Another possible operation for getting upper or lower bounding networks is reducing or increasing the link capacities. As a special case of such operations, one can reduce the capacity of a link to zero which is the same as deleting the link. In some cases reducing/increasing link capacities is helpful in simplifying the network.

Another type of operation for simplifying networks is based on network cut-sets. A cut P between two sets of nodes \mathcal{W}_1 and \mathcal{W}_2 is a partition of the network nodes \mathcal{V} into two sets \mathcal{V}_{c_1} and \mathcal{V}_2 such that $\mathcal{W}_1 \subseteq \mathcal{V}_1$, $\mathcal{W}_2 \subseteq \mathcal{V}_2$. The capacity of a cut is defined as the sum of capacities of the forward links of

the cut, i.e. links (v, w) such that $v \in \mathcal{V}_1, w \in \mathcal{V}_2$. Links (v, w) such that $v \in \mathcal{V}_2, w \in \mathcal{V}_1$ are called backward links of the cut. For example, if we find a minimum cut separating a sink from its corresponding sources and all other sink nodes, and connect the forward links directly to the sink node while preserving all the backward links, this results in an upper-bounding network. If instead of keeping the backward links, we delete them, a lower-bounding network is obtained. In the case where there are no backward links, this procedure results in an equivalent network. We can of course repeat this procedure for every sink.

Another simplification operation involves removing a set \mathcal{A} of links or components and possibly replacing it with additional capacity that might be spread over the remaining network. A simple lower bounding network can be obtained by removing a set \mathcal{A} , while an upper bounding network can be obtained from replacing a set \mathcal{A} by adding sufficient capacity to the remaining network to be able to carry any information that could have been carried by the set \mathcal{A} . For example, if the remaining network contains paths from the inputs to the outputs of the set \mathcal{A} , we can formulate a linear program, based on generalized flows, to find the minimum factor k by which scaling up the capacities of the remaining links uniformly gives an upper bounding network¹. Since the lower bounding network obtained by just removing the set \mathcal{A} differs from the upper bounding network by the scaling factor k , this gives a multiplicative bound of $k - 1$ on the potential capacity difference associated with the upper and lower bounding operations.

V. Y-NETWORKS AND GENERALIZATIONS

Consider the network shown in Fig. 1 consisting of four nodes and three directed links with capacities (r_1, r_2, r_3) . This topology is probably the simplest network in which nodes are sharing resources to send their information. We will refer to such a network as a Y-network $Y(r_1, r_2, r_3)$.

Now consider the network shown in Fig. 2 which consists of two Y-networks with shared input and output nodes. The following lemma shows that in some special cases this network is equivalent to another Y-network. This simplification reduces the number of links by 3 and the number of nodes by 1.

Lemma 1: Consider the network shown in Fig. 2, \mathcal{N}_1 , when $\tilde{a} = \alpha a$, $\tilde{b} = \alpha b$ and $\tilde{c} = \alpha c$. This network is equivalent to \mathcal{N}_2 , a Y-network $Y((1 + \alpha)a, (1 + \alpha)b, (1 + \alpha)c)$.

Proof: Clearly, a Y-network $Y((1 + \alpha)a, (1 + \alpha)b, (1 + \alpha)c)$ is an outer bound for the network of Fig. 2. Hence, we only need to show that it also serves as an inner bound.

For the rest of the proof assume that α is a rational number (If it is not rational, it can be approximated by rational numbers with arbitrary precision).

Consider a code of block length n that runs on network \mathcal{N}_2 . The middle node maps the $n(1 + \alpha)a$ bits received from x_1

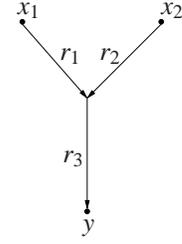


Fig. 1. A Y-network

and the $n(1 + \alpha)b$ bits received from x_2 to $n(1 + \alpha)c$ bits sent to node y .

In order to run the same code on \mathcal{N}_1 , consider using the network m times, where m is chosen such that $\alpha m / (1 + \alpha)n$ and $m / (1 + \alpha)n$ are both integers. Note that since α is rational by assumption, it is always possible to find such m . Let $k_1 \triangleq \alpha m / (1 + \alpha)n$ and $k_2 \triangleq m / (1 + \alpha)n$.

During these m channel uses the intersection node at the left hand side receives ma bits from x_1 and mb bits from x_2 . This is equal to the bits received by the intersection node in \mathcal{N}_2 during $ma / (1 + \alpha)an = k_2$ coding sessions. Therefore, using the code used on \mathcal{N}_2 , these bits can be mapped into $k_2 n(1 + \alpha)c = mc$ bits that will be sent to y . Similarly, the number of bits received by the intersection node on the right hand side during m channel uses is equal to the bits that would have been received by the intersection node on \mathcal{N}_2 during k_1 coding sessions. ■

Lemma 1 serves as a useful tool in our network simplifications. For an example of how to employ this result, consider the network shown in Fig. 3. This network can be considered as a combination of two overlapping Y-networks.

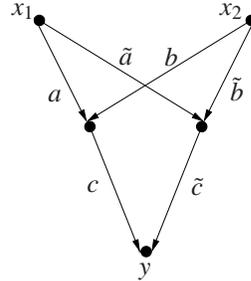


Fig. 2. 2 Y-networks with shared source and sink nodes and separate relay links

Lemma 2: Let $\beta \triangleq \frac{b'}{b+b'}$. If $\beta a + (1 - \beta)c \leq d$, then the network shown in Fig. 3 is equivalent to a Y-network $Y(a, b + b', c)$.

Proof: Clearly a Y-network $Y(a, b + b', c)$ is an upper-bounding network for the network of Fig. 3. We show that if the constraints in the lemma is satisfied, it also serves as a lower-bounding network.

To find a lower bounding network, consider breaking the links in Fig. 3 into parallel links as in Fig. 4, where $a = a_1 + a_2 + a_3$, $b = b_1 + b_2 + b_3$, etc. The network contains two

¹One way to think of this is to associate a commodity with each link or path segment in \mathcal{A} . Conversions between commodities take place according to the link capacity ratios in \mathcal{A} . The linear program minimizes k subject to flow conservation of these commodities.

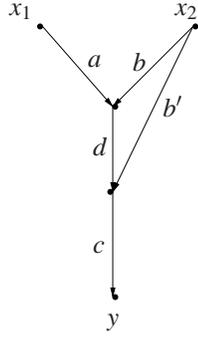


Fig. 3. Two nodes communicating with one sink node via two relay nodes

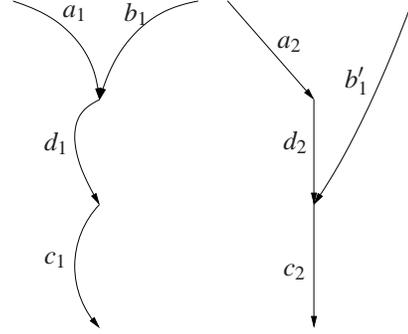


Fig. 5. Breaking the network in Fig. 3 into two Y-networks

capacity-disjoint Y-networks as illustrated in Fig. 5. Our goal is to combine these two Y-networks by applying Lemma 1, and in order to be able to do this, we require $a_2 = d_2 = \alpha a_1$, $b'_1 = \alpha b_1$ and $c_1 = d_1 = \frac{c_2}{\alpha}$. The combination of these two Y-networks will be a Y-network $Y((1 + \alpha)a_1, (1 + \alpha)b_1, (1 + \alpha)c_1)$ which is a lower-bounding network for our original network. Now choosing $b_1 = b$, $b'_1 = b'$ and $\alpha = \frac{b'}{b}$ from the link capacities constraints, we should have

$$\begin{aligned} (1 + \frac{b'}{b})a_1 &\leq a, \\ (1 + \frac{b'}{b})c_1 &\leq c, \\ \frac{b'}{b}a_1 + c_1 &\leq d. \end{aligned} \quad (8)$$

From these inequalities, if $\beta a + (1 - \beta)c \leq d$, we can choose $a_1 = (1 - \beta)a$ and $c_1 = (1 - \beta)c$ and get a lower-bounding Y-network $Y(a, b + b', c)$.

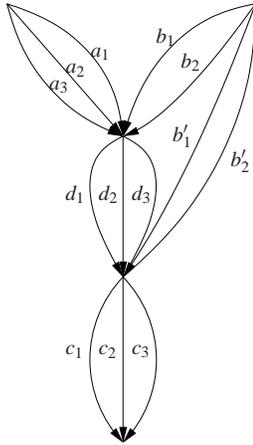


Fig. 4. Breaking the links in Fig. 3 into parallel links

In order to get a better understanding of the required constraint stated in Lemma 2, consider the following special cases:

- i. $b \ll b' \rightarrow d \geq a$,
- ii. $b = b' \rightarrow d \geq \frac{a+c}{2}$,
- iii. $b \gg b' \rightarrow d \geq c$.

Again consider the network shown in Fig. 3 where all the links have capacity 1 except for the link of capacity d . From Lemma 2, for $d \geq 0.5 \times 1 + 0.5 \times 1 = 1$, this network is equivalent to a Y-network $T(1, 2, 1)$. Our proof approach in Lemma 2 does not say anything about the case where $d < 1$. It might be the case that even for some values $d < 1$, still this equivalence holds. As we show through an example this cannot happen, and for $d < 1$ the Y-network of $Y(1, 2, 1)$ is a strict upper bound for our network. Assume that b_1 and (b_2, b_3) are available at x_1 and x_2 respectively, where $b_i \in \{0, 1\}$. The goal is reconstructing $b_1 b_2 + \bar{b}_1 \bar{b}_2 + b_3$ at node y , where all operations are in $GF(2)$. This can be done easily in the Y-network $Y(1, 2, 1)$, but is impossible in the original network for $d < 1$.

Fig. 6 shows a generalization of the network of Fig. 3, where instead of 2 intermediate nodes, there are $k + 1$ intermediate nodes. Let $\alpha_0 \triangleq 1$, $\alpha_i \triangleq b_i/b_0$, for $i \in \{1, \dots, k\}$,

$$a_1 \triangleq \frac{a}{1 + \sum_{i=1}^k \alpha_i},$$

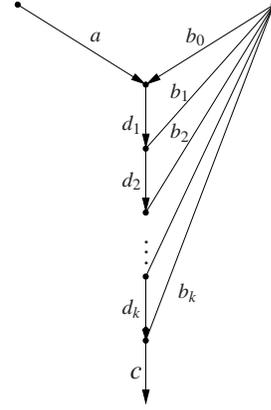


Fig. 6. Generalization of the network of Fig. 3

and

$$c_1 \triangleq \frac{c}{1 + \sum_{i=1}^k \alpha_i}.$$

By extending the proof of Lemma 2 to this more general case, we get Lemma 3.

Lemma 3: If, for $i \in \{1, \dots, k\}$,

$$\sum_{j=0}^{i-1} \alpha_j c_1 + \sum_{j=i}^k \alpha_j a_1 \leq d_i, \quad (9)$$

then the network shown in Fig. 6 is equivalent to a Y-network $Y(a, \sum_{i=0}^k b_i, c)$.

Another possible generalization of a Y-network is shown in Fig. 7. Here, while the number of relay nodes is kept as two, the number of source nodes is increased. For this network, we can prove Lemma 4 with straightforward extension of Lemma 2.

Lemma 4: For $i = \{2, \dots, k\}$, let

$$\beta_i \triangleq \frac{b_i}{b'_i}.$$

Then, if

$$d \geq \sum_{i=1}^n \beta_i a + \sum_{i=1}^n (1 - \beta_i) c, \quad (10)$$

the two intermediate nodes can be combined without changing the performance.

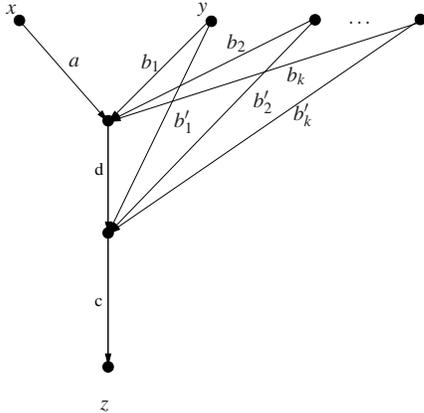


Fig. 7. Another generalization of the network of Fig. 3

Again consider the network shown in Fig. 3, and assume that $d < \beta a + (1 - \beta)c$. As shown in the previous section, a Y-network $Y(a, b + b', c)$, \mathcal{N}_u , serves as a strict upper bound for the network of Fig. 3. In order to get a lower bounding Y-network, we reduce the capacities of the links a and c by a factor δ , such that $d = \beta \delta a + (1 - \beta) \delta c$. Hence,

$$\delta = \frac{d}{\beta a + (1 - \beta) c},$$

and from our assumption $0 < \delta < 1$. Using this, we can again invoke Lemma 2 and get a lower-bounding Y-network of

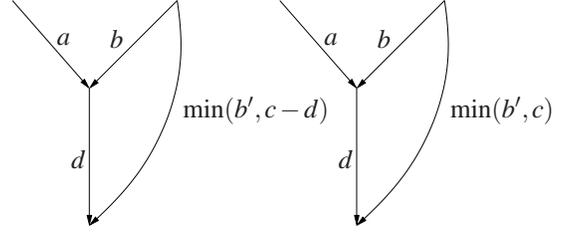


Fig. 8. A pair of upper and lower bounding networks for the network of Fig. 3.

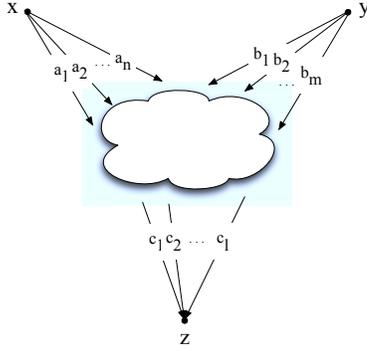
$(\delta a, b + b', \delta c)$ denoted by \mathcal{N}_l . Comparing the link capacities of the upper and lower bounding Y-networks, their difference factor is $\Delta(\mathcal{N}_l, \mathcal{N}_u) = \delta$. If $d \ll \beta a + (1 - \beta)c$, then $\delta \ll 1$, meaning that the bounds become very loose in such cases.

To solve this problem, consider another possible pair of upper and lower bounding networks shown in Fig. 8, which have a topology different than a Y-network. Here the assumption is that $d < c$. It is easy to check that these are indeed upper and lower bounding networks. For deriving the upper-bound, the incoming information from links a and b are assumed to be transmitted directly to the final node, and since the incoming capacity of the final node is c , the information sent to it from link b' can be captured by a direct link of capacity $\min(b', c)$. For the lower bound, since $d < c$, all the information on link c can be directed to the final node. By this strategy, the remaining unused capacity of link c is $c - d$ which can be dedicated to link b' . If $\min(c - d, b') = b'$, then $c \leq d + b'$, then the upper and lower bounding networks coincide, and we have an equivalent network. The more interesting case is when $c > d + b'$, and therefore $\min(c - d, b') = c - d$. In this case the difference factor of the upper and lower bounding networks is at least $1 - d/c$. Choosing the best bound depending on the link capacities, given a, b, b' and c the worst difference factor is

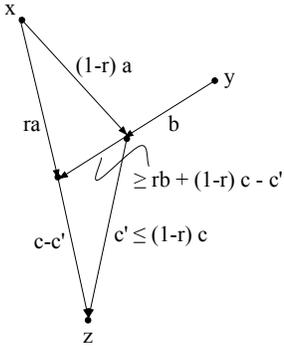
$$\min \max \left\{ \frac{d}{\beta a + (1 - \beta) c}, 1 - \frac{d}{c} \right\} = \frac{c}{c + \beta a + (1 - \beta) c}. \quad (11)$$

As an example, for the case where $a = b = b' = c = 1$, the worst case difference factor is 0.5 which corresponds to $d = 0.5$. This means that choosing the best pair of bounds for different values of d , the difference factor of our selected pair is always lower bounded by 0.5.

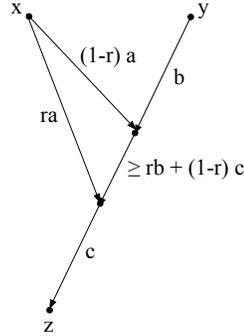
We conclude this section by discussing how the approach in the section generalizes to a larger class of network topologies. Consider the network \mathcal{N}_1 shown in Fig. 9. A Y-network $Y(\sum a_i, \sum b_i, \sum c_i)$ is always an upper bound for the network \mathcal{N}_1 . On the other hand, if we can show, as in the proof of Lemma 2, that \mathcal{N}_1 contains two capacity-disjoint Y-networks of the form $Y(\alpha \sum a_i, \alpha \sum b_i, \alpha \sum c_i)$ and $Y((1 - \alpha) \sum a_i, (1 - \alpha) \sum b_i, (1 - \alpha) \sum c_i)$, then $Y(\sum a_i, \sum b_i, \sum c_i)$ is also a lower bound for, and thus equivalent to, \mathcal{N}_1 . Subfigures (b) and (c) provide two simple examples of networks where this is the case. This construction can also be generalized by replacing the basic Y-shaped topology with star-shaped topologies with arbitrary numbers of inputs and outputs.



(a) Network \mathcal{N}_1



(b) Example 1



(c) Example 2

Fig. 9. A family of components that can be replaced with an equivalent component. The labels represent the link capacities, which satisfy $a = a_1 + a_2 + \dots + a_n$, $b = b_1 + b_2 + \dots + b_m$, $c = c_1 + c_2 + \dots + c_l$, and $0 < r < 1$.

VI. EFFECT OF LINK CAPACITIES

Consider two networks \mathcal{N}_1 and \mathcal{N}_2 which have identical topologies and link capacities except for some link e which has capacity $C_{e,1}$ in \mathcal{N}_1 and capacity $C_{e,2} < C_{e,1}$ in \mathcal{N}_2 . Let \mathcal{R}_1 and \mathcal{R}_2 denote the set of achievable rates on \mathcal{N}_1 and \mathcal{N}_2 respectively. The question is how this difference affects the performances of these two networks. One way of doing this comparison is based on what was mentioned earlier, i.e., to compute the ratio between $C_{e,1}$ and $C_{e,2}$. However, this might not always give the best possible bound. The reason is that it might be case that while $C_{e,1}$ and $C_{e,2}$ are both small compared to the capacity region of the networks, $\Delta(\mathcal{N}_1, \mathcal{N}_2) = C_{e,1}/C_{e,2}$ is very large. In this section, we study this problem in more details.

Note that the link of capacity $C_{e,1}$ in network \mathcal{N}_1 into two parallel links of capacities $C_{e,1} - C_{e,2}$ and $C_{e,2}$. Clearly this process does not affect the capacity region of \mathcal{N}_1 . By this transformation, network \mathcal{N}_2 is equivalent to this new network with link of capacity $\varepsilon \triangleq C_{e,1} - C_{e,2}$ being removed. Therefore, in the rest of this section, instead of changing the capacity of a link, we assume that a link of capacity ε is removed as shown in Fig. 10, and prove that at least in some cases changing the capacity of a link by ε cannot have an effect larger than ε on the set of achievable rates, i.e., if the rate vector \mathbf{R} is achievable

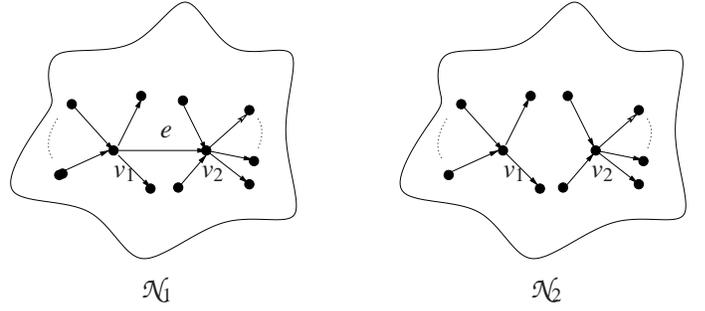


Fig. 10. Networks \mathcal{N}_1 and \mathcal{N}_2 which are with and without link e respectively.

on \mathcal{N}_1 , rate vector $\mathbf{R} - \varepsilon \mathbf{1}$, where $\mathbf{1}$ denotes an all-one vector of length $|\mathcal{S}|$, is achievable on \mathcal{N}_2 as well.

One such example is the case of multicast networks. In that case, the capacity of the network is determined by the values of the cuts from the sources to the sinks. Therefore, since removing a link of capacity ε does not change the values of the cuts by more than ε , the capacity of the network will not be affected by more than ε .

Another example, is the case where all sources are connected directly by a link to a super-source node which has therefore access to the information of all sources. Without loss of generality, let $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$. As before, assume that each sink node $t \in \mathcal{T}$ is interested in recovering a subset of sources denoted by $\beta(t)$.

Theorem 1: For the described network with a super source-node and arbitrary demands, removing a link of capacity ε can change the capacity region by at most ε .

Proof:

Assume that the rate vector $\mathbf{R} = (R_1, R_2, \dots, R_{|\mathcal{S}|})$ is achievable on \mathcal{N}_1 . Since in the single source case, capacity regions corresponding to zero and asymptotically zero probability of errors coincide [12], we can assume there exist a coding scheme of block length n that achieves rate \mathbf{R} on \mathcal{N}_1 with zero probability of error.

Based on this zero-error code, we construct another code for network \mathcal{N}_2 that achieves rate $\mathbf{R} - \varepsilon \mathbf{1}$ with asymptotically zero probability of error.

By our assumption about the network structure, the super source-node observes a message vector $\mathbf{M} = (M_1, \dots, M_{|\mathcal{S}|})$, where $M_s \in \{1, 2, \dots, 2^{nR_s}\}$, for $s \in \{1, 2, \dots, |\mathcal{S}|\}$.

Now consider the link $e = (v_1, v_2)$ in network \mathcal{N}_1 which has been removed in network \mathcal{N}_2 . During the coding process on \mathcal{N}_1 , the bits sent across this link can take on at most $2^{n\varepsilon}$ different values. Consider binning the message vectors $\mathbf{M} = (M_1, M_2, \dots, M_{|\mathcal{S}|})$ into $2^{n\varepsilon}$ different bins based on the bit stream sent over this link during their transmission. Since the code is a deterministic code, each message vector only corresponds to one bin. Since there exist $2^{n\sum_{s \in \mathcal{S}} R_s}$ distinct message vectors, based on the Pigeonhole principle, there will be at least one bin with more than $2^{n(\sum_{s \in \mathcal{S}} R_s - \varepsilon)}$ message vectors. Denote the message vectors contained in this bin by the set \mathcal{M}_0 .

In \mathcal{N}_2 , no message can be sent from v_1 to v_2 through link e .

Therefore, in order to run the code for \mathcal{N}_1 on network \mathcal{N}_2 , we need to specify the assumption of node v_2 about the message that it would have received from node v_1 in \mathcal{N}_1 . Let node v_2 assume that it always receives the bit pattern corresponding to the bin containing \mathcal{M}_0 . Making this assumption and having the rest of \mathcal{N}_2 to perform as in \mathcal{N}_1 , it is clear that all message vectors in \mathcal{M}_0 can be delivered with zero probability of error on \mathcal{N}_2 as well. In other words, if the input to the super source node in \mathcal{N}_2 is one of the message vectors in \mathcal{M}_0 , then each sink $t \in \mathcal{T}$ recovers its intended message $\mathbf{M}_{\beta(t)}$ with probability one. In the rest of the proof we show how this set \mathcal{M}_0 can be used to deliver independent information to different receivers over network \mathcal{N}_2 .

Define random vector $\mathbf{U} = (U_1, U_2, \dots, U_{|\mathcal{S}|})$ to have a uniform distribution over the elements of \mathcal{M}_0 . For each input vector \mathbf{U} , each sink node t , recovers $\mathbf{U}_{\beta(t)}$ perfectly.

The described model with input \mathbf{U} , and outputs $\mathbf{U}_{\beta(t)}$, for $t \in \mathcal{T}$ is a deterministic broadcast channel (DBC)² whose capacity region is known. Therefore, we can employ this DBC to deliver information on \mathcal{N}_2 .

Before doing this, we slightly change the set of sink nodes, and replace the set of sinks \mathcal{T} by \mathcal{T}_e as described in the following. This modification does not affect the functionality of the network, but simplifies the statement of the proof. Divide each sink node t into $|\beta(t)|$ sink nodes, such that each one has access to all the information available to the node t , but is only interested in reconstructing one of the sources. Let \mathcal{T}_e denote this expanded set of sinks. Consider a subset \mathcal{T}_s of size $|\mathcal{S}|$ of \mathcal{T}_e such that each source $s \in \mathcal{S}$ is recovered by one of the sinks in \mathcal{T}_s . Since each sink in \mathcal{T}_s only recovers one source, hence there should be a one-to-one correspondence between the elements of \mathcal{S} and \mathcal{T}_s .

Now consider the DBC with input \mathbf{U} and outputs $\{U_t\}_{t \in \mathcal{T}_s}$. Since the code is zero-error, and there is a one-to-one correspondence between the elements of \mathcal{S} and \mathcal{T}_s , $\{U_s\}_{s \in \mathcal{S}}$ can be replaced by $\{U_t\}_{t \in \mathcal{T}_s}$. The capacity region of this DBC, as explained in Appendix A, can be described by the set of rates $(r_1, r_2, \dots, r_{|\mathcal{S}|})$ satisfying

$$\sum_{s \in \mathcal{A}} r_s \leq H(U_{\mathcal{A}}), \quad (12)$$

for all $\mathcal{A} \subset \mathcal{S}$. From our construction,

$$H(U_{\mathcal{S}}) = H(U_1, U_2, \dots, U_{|\mathcal{S}|}) \\ = n \left(\sum_{s \in \mathcal{S}} R_s - \varepsilon \right). \quad (13)$$

On the other hand, for each $\mathcal{A} \subset \mathcal{S}$

$$H(U_{\mathcal{S}}) = H(U_{\mathcal{A}}) + H(U_{\mathcal{S} \setminus \mathcal{A}}). \quad (14)$$

²DBCs and their capacity regions are briefly described in Appendix A.

Therefore,

$$H(U_{\mathcal{A}}) = H(U_{\mathcal{S}}) - H(U_{\mathcal{S} \setminus \mathcal{A}}) \\ \geq n \left(\sum_{s \in \mathcal{S}} R_s - \varepsilon \right) - \sum_{s \in \mathcal{S} \setminus \mathcal{A}} H(U_s) \\ \geq n \left(\sum_{s \in \mathcal{S}} R_s - \varepsilon \right) - \sum_{s \in \mathcal{S} \setminus \mathcal{A}} n R_s \\ = n \left(\sum_{s \in \mathcal{A}} R_s - \varepsilon \right). \quad (15)$$

It is easy to check that the point $(r_1, r_2, \dots, r_{|\mathcal{S}|}) = n(\mathbf{R} - \varepsilon \mathbf{1})$ satisfies all inequalities required by (12). Hence, using the network \mathcal{N}_2 as a DBC, the rate vector $\mathbf{R} - \varepsilon$ can be transmitted to nodes in \mathcal{T}_s .

We now argue that the described DBC code, with no extra effort, delivers the rate vector $\mathbf{R} - \varepsilon$ on network \mathcal{N}_2 to all sinks, not just those in \mathcal{T}_s . The reason is that for each message vector $\mathbf{M} \in \mathcal{M}_0$, and consequently for each input \mathbf{U} , all sink nodes in \mathcal{T}_e that are interested in recovering a source s , i.e., for all $t \in \mathcal{T}_e$ such that $\beta(t) = s$, receive M_s with probability one. Hence, if by the described coding scheme rate R_s is delivered to one of them, then the rest are able to receive the same data stream at rate R_s as well. ■

For some simple cases of multi-source multi-demand case networks, for instance when the link of capacity ε is directly connected to one of the sources, the same result still holds, i.e. removing that link cannot change the capacity region by more than ε . However, for the general case, it remains open as to how much removing a link of capacity ε can affect the capacity region of the network.

VII. CYCLES AND DELAYS

In [13], the capacity of a channel is defined in terms of bits per channel use. In the case where there are no cycles, this definition is equivalent to defining capacity in terms of bits transmitted per unit time. However, we do not know whether in a general network with cycles and delays, these two definitions, i.e., bits per channel use versus bits per unit time, could result in different capacity regions. For example, consider two networks \mathcal{N}_1 and \mathcal{N}_2 that are identical except for a single component cycle, illustrated in Fig. 11. In \mathcal{N}_2 the transmission delay from v_1 to v_2 is larger than in \mathcal{N}_1 due to the added node v_0 , assuming that each link introduces unit delay. Now, we do not know whether achieving capacity on \mathcal{N}_1 could require interactive communication over a feedback loop, where each successive symbol transmitted by v_1 depends on previous symbols received from v_2 and vice versa. If we try to run the same code on \mathcal{N}_2 , the number of symbols per unit time that can be exchanged by nodes v_1 and v_2 is decreased because of the additional delay introduced by node v_0 . Hence, if the rate is defined in terms of bits communicated per unit time, this additional delay will reduce the capacity region. However, if the capacity is defined in terms of bits communicated per channel use, by letting each node wait until it has received the information it requires to generate its next symbol, the

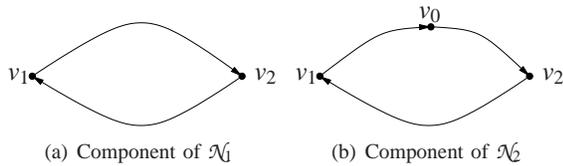


Fig. 11. Changing the delay between two nodes by introducing an extra node

delivered rates and consequently the capacity region will not be affected.³

In most of the operations introduced in this paper, the number of hops between nodes is different in the original network compared to its simplified version. Therefore, such operations in a cyclic graph with delay can potentially change the capacity region in ways that are not predicted by our analysis. Hence, in this paper we have restricted ourselves to the case of acyclic graphs. However, studying the effect of these operations in a general network with cycles remains as a question for further study.

VIII. CONCLUSIONS

In this paper, we proposed new techniques for efficiently approximating or bounding the capacity of networks with complex topologies and arbitrary communication demands, such as non-multicast and functional demands. We proposed to take a new approach based on systematic graph operations that can be applied recursively to simplify large networks, so as to facilitate the application of computational bounding tools. Besides its generality, another motivation for such a recursive graph-based approach is that it lends itself to tractable “divide-and-conquer” algorithms for analyzing very large networks and allows computation to be traded off against tightness of obtained bounds according to the extent of simplification.

Techniques proposed for networks of directed noiseless point-to-point links (bit pipes) can be readily applied to networks of noisy multi-terminal channels by using the results of Koetter et al. [9] to replace each noisy link with a bounding model consisting of such bit pipes. However, it may be possible to obtain better bounds by developing simplification tools that are directly applicable to noisy networks. Thus, we will seek to extend the above work to networks containing noisy broadcast, multiple access and interference links.

APPENDIX A

DETERMINISTIC BROADCAST CHANNEL

Deterministic broadcast channels (DBC) are special cases of general broadcast channels. In a K -user DBC,

$$P((Y_1, \dots, Y_K) = (y_1, \dots, y_K) | X = x) \in \{0, 1\}.$$

In other words, since in a BC the capacity region only depends on the marginal distributions [13], a K -user DBC can be

³Note that in the case of multicast, since it is known that capacity-achieving codes do not require such communication over feedback loops, the capacity region remains unchanged under both definitions, even in the case of cyclic graphs.

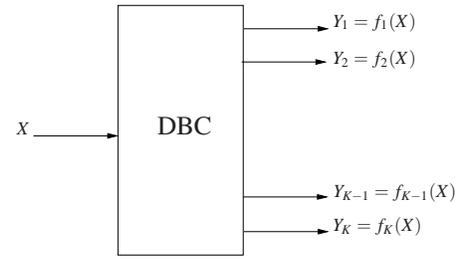


Fig. 12. Deterministic broadcast channel

described by k functions (f_1, \dots, f_k) where

$$f_i : \mathcal{X} \rightarrow \mathcal{Y}_i, \quad (\text{A-1})$$

and $Y_i = f_i(X)$ for $i \in \{1, 2, \dots, k\}$. In (A-1), \mathcal{X} and \mathcal{Y}_i refer to the channel input alphabet and the output alphabet of the i^{th} channel respectively.

The capacity region of a k -user DBC can be described by the union of the set of rates (R_1, R_2, \dots, R_k) satisfying

$$\sum_{i \in \mathcal{A}} R_i \leq H(Y_{\mathcal{A}}), \forall \mathcal{A} \subset \{1, \dots, k\}, \quad (\text{A-2})$$

for some $P(X)$ [14], [15].

ACKNOWLEDGMENTS

This work has been supported in part by the Air Force Office of Scientific Research under grant FA9550-10-1-0166 and Caltech’s Lee Center for Advanced Networking.

REFERENCES

- [1] Kramer. G. and S. Savari. Capacity bounds for relay networks. In *Proc. 2006 Workshop Inf. Theory Appl.*, La Jolla, CA, feb 2006.
- [2] M. Adler, N. J. Harvey, K. Jain, R. Kleinberg, and A. R. Lehman. On the capacity of information networks. In *SODA*, 2006.
- [3] N. J. A. Harvey, R. Kleinberg, C. Nair, and Y. Wu. A “chicken and egg” network coding problem. In *Proc. IEEE Int. Symp. Inform. Theory*, Nice, France, June 2007.
- [4] R.W. Yeung. A framework for linear information inequalities. *Information Theory, IEEE Transactions on*, 43(6):1924–1934, nov. 1997.
- [5] A.T. Subramanian and A. Thangaraj. A simple algebraic formulation for the scalar linear network coding problem. *ArXiv e-prints*, July 2008.
- [6] Xitip - information theoretic inequalities prover. <http://xitip.epfl.ch/>.
- [7] M. Médard and R. Koetter. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, pages 782–795, 2003.
- [8] T. H. Chan. Capacity regions for linear and abelian network code. In *NETCOD*, San Diego, CA, January 2007.
- [9] R. Koetter, M. Effros, and M. Médard. A theory of network equivalence, parts i and ii. *ArXiv e-prints*, July 2010.
- [10] R. W. Yeung. *Information Theory and Network Coding*. Springer, 2008.
- [11] M. Effros. On capacity outer bounds for a simple family of wireless networks. In *Information Theory and Applications Workshop*, San Diego, CA, 2010.
- [12] T. Chan and A. Grant. On capacity regions of non-multicast networks. In *Proc. IEEE Int. Symp. Inform. Theory*, Austin, TX, June 2010.
- [13] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 2nd edition, 2006.
- [14] K. Marton. The capacity region of deterministic broadcast channels. In *Proc. IEEE Int. Symp. Inform. Theory*, Paris-Cachan, France, 1977.
- [15] M. S. Pinsker. Capacity of noiseless broadcast channels. *Probl. Inform. Transm.*, pages 92–102, 1978.