Communicating the sum of sources over a network

Aditya Ramamoorthy
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011
Email: adityar@iastate.edu

***Abstract*— We consider a network (that is capable of network coding) with a set of sources and terminals, where each terminal is interested in recovering the sum of the sources. Considering directed acyclic graphs with unit capacity edges and independent, unit-entropy sources, we show the rate region when (a) there are two sources and n terminals, and (b) n sources and two terminals. In these cases as long as there exists at least one path from each source to each terminal we demonstrate that there exists a valid assignment of coding vectors to the edges such that the terminals can recover the sum of the sources.**

I. Introduction

Network coding is a new paradigm in networking where nodes in a network have the ability to process information before forwarding it. This is unlike routing where nodes in a network primarily operate in a replicate and forward manner. The problem of multicast has been studied intensively under the paradigm of network coding. The seminal work of Ahlswede et al. [1] showed that under network coding the multicast capacity is the minimum of the maximum flows from the source to each individual terminal node. The work of Li et al. [2] showed that linear network codes were sufficient to achieve the multicast capacity. The algebraic approach to network coding proposed by Koetter and Médard [3] provided simpler proofs of these results.

In recent years there has also been a lot of interest in the development and usage of distributed source coding schemes due to their applications in emerging areas such as sensor networks. Classical distributed source coding results such as the famous Slepian-Wolf theorem [4] usually assume a direct link between the sources and the terminals. However in applications such as sensor networks, typically the sources would communicate with the terminal over a network. Thus, considering the distributed compression jointly with the network information transfer is important. Network coding for correlated sources was first examined by Ho et al. [5]. The work of Ramamoorthy et al. [6] showed that in general separating distributed source coding and network coding is suboptimal except in the case of two sources and two terminals. A practical approach to transmitting correlated sources over a network was considered by Wu et al. [7]. Reference [7] also introduced the problem of *Network Arithmetic* that comes up in the design of practical systems that combine distributed source coding and network coding.

In the network arithmetic problem, there are source nodes each of which is observing independent sources. In addition there is a set of terminal nodes that are only interested in the sum of these sources i.e. unlike the multicast scenario where the terminals are actually interested in recovering all the sources, in this case the terminals are only interested in the sum of the sources. In this paper we study the rate region of the network arithmetic problem under certain special cases. In particular we restrict our attention to directed acyclic graphs (DAGs) with unit capacity edges and independent, unit entropy sources. Moreover, we consider the following two cases.

i) Networks with two sources and n terminals, and
ii) networks with n sources and two terminals.

For these two cases we present the rate region for the problem. Basically we show that as long as there exists at least one path from each source to each terminal, there exists an assignment of coding vectors to each edge in the network such that the terminals can recover the sum of the sources.

This paper is organized as follows. Section II presents the network coding model that we shall be assuming. Section III contains our results for the case when there are two sources and n terminals and section IV contains the results and proofs for the case when there are n sources and two terminals. In section V we outline our conclusions.

II. Network coding model

In our model, we represent the network as a directed graph $G = (V, E)$. The network contains a set of source nodes $S \subset V$ that are observing independent, discrete unit-entropy sources and a set of terminals $T \subset V$. Our network coding model is basically the one presented in [3]. We assume that each edge in the network has unit capacity and can transmit one symbol from a finite field of size 2^m per unit time (we are free to choose m large enough). If a given edge has a higher capacity, it can be treated as multiple unit capacity edges (fractional capacities can be treated by choosing m large enough). A directed edge e between nodes v_i and v_j is represented as $(v_i \to v_j)$. Thus $head(e) = v_j$ and $tail(e) = v_i$. A path between two nodes v_i and v_j is a sequence of edges $\{e_1, e_2, \dots, e_k\}$ such that $tail(e_1) = v_i, head(e_k) = v_j$ and $head(e_i) = tail(e_{i+1}), i = 1, \dots k-1$.

The signal on an edge $(v_i \to v_j)$, is a linear combination of the signals on the incoming edges on v_i and the source signal at v_i (if $v_i \in S$). In this paper we assume that the source nodes do not have any incoming edges from other nodes. If this is not the case one can always introduce an artificial source connected to the original source node that has no incoming edges. We shall only be concerned with

This research was supported in part by NSF grant CNS-0721453.

networks that are directed acyclic and can therefore be treated as delay-free networks [3]. Let Y_{e_i} (such that $tail(e_i) = v_k$ and $head(e_i) = v_l$) denote the signal on the i^{th} edge in E and let X_j denote the j^{th} source. Then, we have

$$Y_{e_i} = \sum_{\{e_j | head(e_j) = v_k\}} f_{j,i} Y_{e_j} \text{ if } v_k \in V \backslash S, \text{ and}$$
$$Y_{e_i} = \sum_{\{j | X_j \text{ observed at } v_k\}} a_{j,i} X_j \text{ if } v_k \in S,$$

where the coefficients $a_{j,i}$ and $f_{j,i}$ are from $GF(2^m)$. Note that since the graph is directed acyclic, it is possible to express Y_{e_i} for an edge e_i in terms of the sources X_j's. Suppose that there are n sources $X_1, \ldots, X_n$. If $Y_{e_i} = \sum_{k=1}^{n} \beta_{e_i,k} X_k$ then we say that the global coding vector of edge e_i is $\boldsymbol{\beta}_{e_i} = [\beta_{e_i,1} \ \cdots \ \beta_{e_i,n}]$. We shall also occasionally use the term coding vector instead of global coding vector in this paper. We say that a node v_i (or edge e_i) is downstream of another node v_j (or edge e_j) if there exists a path from v_j (or e_j) to v_i (or e_i).

III. Case of two sources and n terminals

In this section we state and prove the rate region for the network arithmetic problem when there are two sources and n terminals.

The basic idea of the proof is the following. We show that there exist a certain set of nodes that can obtain both the sources X_1 and X_2 and find a multicast code that multicasts the pair (X_1, X_2) to these nodes. We then modify the set of coding vectors so that all the terminals can recover $X_1 + X_2$ while ensuring that the coding vectors remain valid.

Theorem 1: Consider a directed acylic graph $G = (V, E)$ with unit capacity edges, two source nodes S_1 and S_2 and n terminal nodes $T_1, \ldots, T_n$ such that

$$\text{max-flow}(S_i - T_j) \geq 1 \text{ for all } i = 1, 2 \text{ and } j = 1, \ldots, n.$$

At each source node S_i, there is a unit-rate source X_i. The X_i's are independent. There exists an assignment of coding vectors to all edges such that each $T_i, i = 1, \ldots, n$ can recover $X_1 + X_2$.

Before embarking on the proof of this result we define a modified graph that shall simplify our later arguments.

1) We introduce artificial source nodes S'_1 and S'_2 such that there exists a unit capacity edge $S'_i \rightarrow S_i$. Similarly we introduce artificial terminal nodes T'_i and unit capacity edges $T_i \rightarrow T'_i$. Note that we are given the existence of at least one path from $S_i \rightarrow T_j$ for all i, j. This in turn implies that max-flow$(S'_1 - T'_j) =$ max-flow$(S'_2 - T'_j) =$ max-flow$((S'_1, S'_2) - T'_j) = 1$.
2) For each virtual terminal $T'_j, j = 1, \ldots, n$ there exists a path from S'_i to T'_j for $i = 1, 2$. Let us denote this by $path(S'_i - T'_j)$. We say that two paths intersect if they have at least one node in common. For a given terminal T'_j, in general the $path(S'_1 - T'_j)$ and $path(S'_2 - T'_j)$ could intersect in many nodes. Note that they have to intersect at least once since the edge $T_j \rightarrow T'_j$ is of unit capacity. Suppose that the first intersection point is denoted v_j. As demonstrated in Fig. 1 it is possible to find a new set of paths from $S'_1 - T'_j$ and $S'_2 - T'_j$ so that they share the set of edges from v_j to T'_j.
We assume that such paths have been found for all terminals. Thus for each terminal T'_j there exists a corresponding v_j which denotes the first vertex where the paths $S'_1 - T'_j$ and $S'_2 - T'_j$ meet. Note that the v_j's may not be distinct. Now, consider the subgraph of G that is defined by the union of all these paths and suppose that we call it G'. In our discussion we shall only be concerned with the graph G'.
3) Note that G' is also a directed acyclic graph. Therefore a numbering of the nodes exists such that if there exists a path between node v_i and v_j then $i < j$. We now number the nodes in G' in this manner. We shall refer to the first meeting point of path$(S'_1 - T'_j)$ and path$(S'_2 - T'_j)$ under this new numbering as $v_{\alpha(T_j)}$.

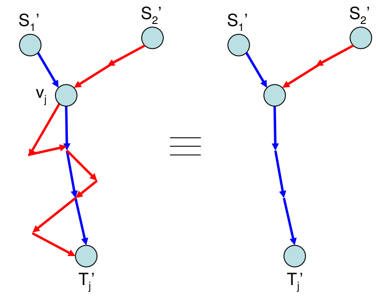

Fig. 1. The figure on the left shows $path(S'_1 - T'_j)$ (in blue) and $path(S'_2 - T'_j)$ (in red). The figure on the right shows that one can find a new set of paths from S'_1 and S'_2 to T'_j such that they share edges from v_j to T'_j. The first intersection of the new paths is at node v_j.

Lemma 1: In the graph G' constructed as above, the following properties hold for all $j = 1, \ldots, n$.

$$\text{max-flow}(S'_1 - v_{\alpha(T_j)}) = 1, \quad (1)$$
$$\text{max-flow}(S'_2 - v_{\alpha(T_j)}) = 1, \text{ and} \quad (2)$$
$$\text{max-flow}((S'_1, S'_2) - v_{\alpha(T_j)}) = 2. \quad (3)$$

Proof. Obvious by the construction of the graph G'. ■

The previous claim implies that there exists a network code so that the pair (X_1, X_2) can be multicast to each node $v_{\alpha(T_j)}, j = 1, \ldots, n$ using Theorem 8 in [3] . Suppose that such a network code is found and the global coding vectors for each edge in G' are found. Let these global coding vectors be specified by the set $\boldsymbol{\beta} = \{\beta_e \mid e \in E'\}$.

We now present an algorithm that modifies $\boldsymbol{\beta}$ so that each terminal $T'_i, i = 1, \ldots, n$ can recover $X_1 + X_2$. This shall serve as a proof of Theorem 1. First we sort the set $\{v_{\alpha(T_1)}, \ldots, v_{\alpha(T_n)}\}$ to obtain $\{v_{\gamma_1}, \ldots, v_{\gamma_n}\}$ so that $\gamma_1 \leq \cdots \leq \gamma_n$. Let the terminal node corresponding to the node v_{γ_i} be denoted $T'_{f(\gamma_i)}$. As mentioned before it is possible that there exist terminals T_i and T_j such that $\alpha(T_i) = \alpha(T_j)$. Therefore the set of γ_i's is not distinct. Consequently the mapping $f(\gamma_i)$ is one to many. We do not make this explicit to avoid the notation becoming too complex. The steps are presented in Algorithm 1.

It is important to note that this algorithm may replace the existing coding vectors assigned by the multicast code construction on some edges. We now show that the new

```
1  Initialize demand[i] = 0, i = 1, ... n;
2  for k ← 1 to n do
3      if demand[f(γ_k)] == 0 then
4          for e ∈ path(v_{γ_k} − T'_{f(γ_k)}) do
5              β_e = [1 1];
6          end
7          demand[f(γ_k)] = 1;
8          for m ← k + 1 to n do
9              if demand[f(γ_m)] == 0 then
10                 if there exists a path(v_{γ_k} − T'_{f(γ_m)})
                   then
11                     for e ∈ path(v_{γ_k} − T'_{f(γ_m)}) do
12                         β_e = [1 1];
13                     end
14                     demand[f(γ_m)] = 1;
15                 end
16             end
17         end
18     end
19 end
```

Algorithm 1: Algorithm for assigning coding vectors so that each terminal can recover the sum of the two sources.

global coding vector assignment is valid and is such that each terminal receives $X_1 + X_2$.

Proof of Theorem 1.

We claim that the assignment of coding vectors is valid at each stage of the algorithm and by stage $1 \leq k \leq n$, demand$[f(\gamma_k)] = 1$.

- *Base case (k=1).* Note that by the construction of G' there exists a path from v_{γ_1} to $T_{f(\gamma_1)}$. The algorithm shall assign coding vector $[1\ 1]$ to those edges and set $demand[f(\gamma_1)] = 1$. We only need to ensure that the assignment is valid. To see the validity of the assignment note that the graph is acyclic, therefore the coding vectors on $path(S'_1 - v_{\gamma_1})$ and $path(S'_2 - v_{\gamma_1})$ do not change. The assignments are only done on edges downstream of v_{γ_1} and are therefore valid.
- *Induction Step.* Assume that the claim is true for all $j = 1, \ldots, k$ and consider stage $k+1$. If for a given j, the algorithm enters the for loop on lines 4-6, we call the node v_{γ_j} an active node.
  1) *Case 1.* If there exists a path between some active node v_{γ_j} in the set $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$ and $T'_{f(\gamma_{k+1})}$ then $demand[f(\gamma_{k+1})]$ will be set to 1 at one of the earlier stages. By the inductive hypothesis, the assignment is valid.
  2) *Case 2.* If $demand[f(\gamma_{k+1})]$ is still zero after k iterations of the algorithm, this implies that there does not exist a path between an active node and $T'_{f(\gamma_{k+1})}$ i.e. there does not exist a path from an active node to any node on $path(S'_1 - T'_{f(\gamma_{k+1})})$ and $path(S'_2 - T'_{f(\gamma_{k+1})})$. Therefore the coding vectors on the edges in $path(S'_1 - T'_{f(\gamma_{k+1})}) \cup path(S'_2 - T'_{f(\gamma_{k+1})})$ are unchanged at the end of iteration k and are such that $v_{\gamma_{k+1}}$ receives (X_1, X_2). This implies that setting $\beta_e = [1\ 1]$ for $e \in path(v_{\gamma_{k+1}} - T_{f(\gamma_{k+1})})$ will ensure that $demand[f(\gamma_{k+1})] = 1$. This assignment is valid since the coding vector [1 1] lies in the span of the coding vector space of $v_{\gamma_{k+1}}$. Furthermore, there does not exist a path from $v_{\gamma_{k+1}}$ to any node on $\bigcup_{j=1}^{k} path(S'_1 - v_{\gamma_j}) \cup path(S'_2 - v_{\gamma_j})$ since the graph is acyclic. Therefore the assignment of coding vectors to the previous edges remains valid. ■

Note that conversely if any of the conditions in the statement of Theorem 1 is violated then there exists some terminal that cannot obtain the value of $X_1 + X_2$. To see this note that since the graph has unit-capacity edges the max-flow between any pair of nodes has to be an integer. Further, if for example max-flow$(S_1 - T_j) = 0$, then the received signal at T_j cannot depend on X_1. Since, X_1 and X_2 are independent, $X_1 + X_2$ cannot be computed at T'_j.

IV. Case of n Sources and Two Terminals

We now present the rate region for the situation when there are n sources and two terminals such that each terminal wants to recover the sum of the sources.

To show the main result we first demonstrate that the original network can be transformed into another network where there exists exactly one path from each source to each terminal. This ensures that when network coding is performed on this transformed graph the gain on the path from a source to a terminal can be specified by a monomial. By a simple argument it then follows that coding vectors can be assigned so that the terminals recover the sum of the sources.

Theorem 2: Consider a directed acylic graph $G = (V, E)$ with unit capacity edges. There are n source nodes $S_1, S_2, \ldots, S_n$ and two terminal nodes T_1 and T_2 such that

$$\text{max-flow}(S_i - T_j) \geq 1 \text{ for all } i = 1, \ldots, n \text{ and } j = 1, 2.$$

At the source nodes there are independent unit-rate sources $X_i, i = 1, \ldots, n$. There exists an assignment of coding vectors such that each terminal can recover the modulo-two sum of the sources $\sum_{i=1}^{n} X_i$.

As before we modify the graph G by introducing virtual source nodes $S'_i, i = 1, \ldots n$, virtual terminals $T'_j, j = 1, 2$ and virtual unit-capacity edges $S'_i \to S_i, i = 1, \ldots, n$ and $T_j \to T'_j, j = 1, 2$. Let the set of sources be denoted $S = \{S'_1, \ldots, S'_n\}$. We denote the modified graph by G'. We also need the following definitions.

Definition 1: Exactly one path condition. Consider two nodes v_1 and v_2 such there is a path $\mathcal{P}$ between v_1 and v_2. We say that there exists exactly one path between v_1 and v_2 if there does not exist another path $\mathcal{P}'$ between v_1 and v_2 such that $\mathcal{P}' \neq \mathcal{P}$.

Definition 2: Minimality. Consider the directed acyclic graph G' defined above, with sources $S'_1, \ldots, S'_n$ and terminals T'_1 and T'_2 such that

$$\text{max-flow}(S'_i - T'_j) = 1 \ \forall\ i = 1, \ldots, n \text{ and } j = 1, 2. \quad (4)$$

The graph G' is said to be minimal if the removal of any edge from E' violates one of the equalities in (4).

To show that Theorem 2 holds we first need an auxiliary lemma that we state and prove.

Lemma 2: Consider the graph G' as constructed above with sources $S'_1, \ldots, S'_n$ and terminals T'_1 and T'_2. There exists a subgraph G^* of G' such that G^* is minimal and there exists exactly one path from S'_i to T'_j for $i = 1, \ldots, n$ and $j = 1, 2$ in G^*.

Proof. We proceed by induction on the number of sources.

- *Base case* $n = 1$. In this case there is only one source S'_1 and both the terminals need to recover X_1. Note that we are given the existence of $path(S'_1 - T'_1)$ and $path(S'_1 - T'_2)$ in G'. In general these paths can intersect at multiple nodes which may imply that there exist multiple paths (for example) from S'_1 to T'_1. Now, from $path(S'_1 - T'_1)$ and $path(S'_1 - T'_2)$ we can find the last node where these two paths meet. Let this last node be denoted u_1. Then as shown in Fig. 2 we can find a new set of paths from S'_1 to T'_1 and S'_1 to T'_2 that overlap from S'_1 to u_1 and have no overlap thereafter. Choose G^* to be the union of these new set of paths. It is easy to see that in G^* there is exactly one path from S'_1 to T'_1 and exactly one path from S'_1 to T'_2. Moreover removing any edge from G^* would cause at least one path to not exist.

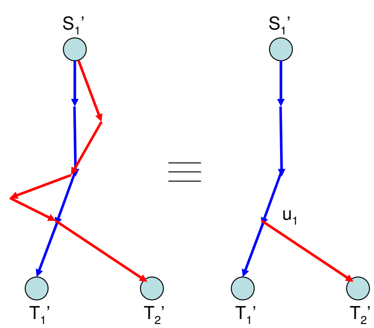

Fig. 2. The figure on the left shows $path(S'_1 - T'_1)$ (in blue) and $path(S'_1 - T'_2)$ (in red). The figure on the right shows that one can find a new set of paths from S'_1 to T'_1 and T'_2 such that they share edges from S'_1 to u_1 and have no intersection thereafter.

- *Induction Step.* We now assume the induction hypothesis for $n - 1$ sources. i.e. there exists a minimal subgraph G^*_{n-1} of G' such that there is exactly one path from S'_i to T'_j for $i = 1, \ldots, n-1$ and $j = 1, 2$. Using this hypothesis we shall show the result in the case when there are n sources.

  As a first step color the edges in the subgraph G^*_{n-1}, blue (the remaining edges in G' have no color). The conditions on G' guarantee the existence of $path(S'_n - T'_1)$ and $path(S'_n - T'_2)$. Note that these paths may intersect at many nodes. We preprocess them in the following manner. Find the last node not in G^*_{n-1} belonging to both $path(S'_n - T'_1)$ and $path(S'_n - T'_2)$. Suppose that this node is denoted v_r. Find a new set of paths such that they share edges from S'_n to v_r and call these new paths $path(S'_n - T'_1)$ and $path(S'_n - T'_2)$. Color all edges on $path(S'_n - T'_1)$ and $path(S'_n - T'_2)$ red. This would imply that some edges have a pair of colors. Now, consider the subgraph induced by the union of the blue and red subgraphs that we denote G_{br}.

  Find the first node at which $path(S'_n - T'_1)$ intersects the blue subgraph and call that node u_1. Similarly find the first node at which $path(S'_n - T'_2)$ intersects the blue subgraph and call that node u_2.

  Observe that in G^*_{n-1} there has to exist a $path(S'_i - T'_j)$ for some $i = 1, \ldots, n-1$ and $j = 1, 2$ that passes through u_1. To see this assume otherwise. This implies that u_1 does not lie on any path connecting one of the sources to one of the terminals. Therefore the incoming and the outgoing edges of u_1 can be removed without violating the max-flow conditions in (4). This contradicts the minimality of G^*_{n-1}. Therefore we are guaranteed that there exists at least one source such that there exists an exclusively blue path from it to u_1 in G^*_{n-1}. A similar statement holds for the node u_2. We now establish the statement of the lemma when there are n sources.

  – *Case 1.* In G_{br} there exists a path from u_1 to T'_2 such that all edges on this path have a blue component.

    First, we remove the color red from all edges on $path(S'_n - T'_2) \backslash path(S'_n - T'_1)$. Next, form a subset of the sources denoted $S^{(u_1)}$ in the following manner. For each source $S'_i, i = 1, \ldots, n$ do the following.

    i) If there exists a path (with edges of color red or blue) from S'_i to u_1, add it to set $S^{(u_1)}$ [1].

    Let $G^{(u_1)}$ denote the subgraph induced by $\bigcup_{S'_i \in S^{(u_1)}} path(S'_i - u_1)$.

    Consider the graph obtained by removing the subgraph $G^{(u_1)}$ from G_{br}. We denote this graph G^-_{br}. We claim that the max-flow conditions in (4) continue to hold over G^-_{br} for the set of sources $S \backslash S^{(u_1)}$. Furthermore there still exist $path(u_1 - T'_1)$ and $path(u_1 - T'_2)$ in G^-_{br}.

    To see this note that the max-flow conditions for a source $S'_i \in S \backslash S^{(u_1)}$ can be violated only if an edge e belonging to a path from S'_i to $T'_j, j = 1, 2$ is removed. This happens only if there exists a path from e to u_1 which contradicts the fact that $S'_i \in S \backslash S^{(u_1)}$. Next, there still exist paths from u_1 to the terminals since the edges on these paths are downstream of u_1. If any of these was removed by the procedure, this would contradict the acyclicity of the graph.

    Note that the subgraph $G^{(u_1)}$ contains a set of sources $S^{(u_1)}$ and a single node u_1 such that there exists exactly one path from each source in $S^{(u_1)}$ to u_1. This has to be true for the sources in $S^{(u_1)} \backslash \{S_n\}$ otherwise the minimality of G^*_{n-1} would be contradicted and is true for S_n by construction.

    Next, introduce an artificial source S_a and an edge $S_a \rightarrow u_1$ in G^-_{br}. Note that $|S \backslash S^{(u_1)}| \leq n - 2$, which means that the total number of sources in G^-_{br} (including S_a) is at most $n - 1$. Therefore the induction hypothesis can be applied on G^-_{br} i.e. there

[1] A path from S'_i to u_1 cannot have a (red,blue) edge since u_1 is the first node where a red path intersects the blue subgraph

exists a subgraph of G_{br}^- such that there exists exactly one path from $(S \backslash S^{(u_1)}) \cup \{S_a\}$ to each terminal. Suppose that we find this subgraph. Now remove S_a and the edge $S_a \rightarrow u_1$ from this subgraph and augment it with the subgraph $G^{(u_1)}$ found earlier. We claim that the resulting graph has the property that there exists exactly one path from each source to each terminal.

To see this note that there exists only one path from a source $S'_i \in S \backslash S^{(u_1)}$ to $T'_j, j = 1, 2$. This is because even after the introduction of $G^{(u_1)}$ there does not exist a path from S'_i to u_1 in this graph. Therefore the introduction of $G^{(u_1)}$ cannot introduce additional paths between $S'_i \in S \backslash S^{(u_1)}$ and the terminals. Next we argue for a source $S'_i \in S^{(u_1)}$. Note that there exists exactly one path from u_1 to both the terminals so the condition can be violated only if there exist multiple paths from $S'_i \in S^{(u_1)}$ to u_1, but the construction of $G^{(u_1)}$ rules this out.

– *Case 2.* In G_{br} there exists a path from u_2 to T'_1 such that all edges on this path have a blue component.

  This case can be handled in exactly the same manner as in case 1 by removing the color red from all edges on $path(S'_n - T'_1) \backslash path(S'_n - T'_2)$ and applying similar arguments for u_2.

– *Case 3.* In G_{br} there (a) does not exist a path with blue edges from u_1 to T'_2, and (b) does not exist a path with blue edges from u_2 to T'_1.

  As shown previously u_1 lies on some path from S'_i to T'_j for some i and j in G^*_{n-1}. In the current case there does not exist a blue path from u_1 to T'_2. Therefore there has to exist a blue path from u_1 to T'_1 in G^*_{n-1}. A similar argument shows that there has to exist a blue path from u_2 to T'_2 in G^*_{n-1}.

  Note that the exclusively red paths from S'_n to u_1 and u_2 are such that they overlap until their last intersection point. Now, choose the desired subgraph to be the union of G^*_{n-1} and the red paths, $path(S'_n - u_1)$ and $path(S'_n - u_2)$ i.e. $G^*_n = G^*_{n-1} \cup path(S'_n - u_1) \cup path(S'_n - u_2)$. By the induction hypothesis there exists exactly one path between $S'_i, i = 1, \ldots, n-1$ and $T'_j, j = 1, 2$. This continues to be true in G^*_n, since the red edges cannot be reached from the blue edges. To see that there is exactly one path from S'_n to T'_1, assume otherwise and observe that there is exactly one path from S'_n to u_1 by the construction of the red paths. Thus the only way there can be multiple paths from S'_n to T'_1 is if there are multiple paths from u_1 to T'_1, but this would contradict the induction hypothesis since this would imply that there exists some $S'_i, i = 1, \ldots, n-1$ that has multiple paths to T'_1. A similar argument shows that there exists exactly one path from S'_n to T'_2. ■

Proof of Theorem 2. From Lemma 2 we know that it is possible to find a subgraph G^* of G such that there exists exactly one path from S'_i to T'_j for all $i = 1, \ldots, n$ and $j = 1, 2$. Suppose that we find G^*. We will show that each terminal can recover $\sum_{i=1} X_i$ by assigning appropriate local encoding responsibilities for every node. Consider a node $v \in G^*$ and let $\Gamma^o(v)$ and $\Gamma^i(v)$ represent the set of outgoing edges from v and incoming edges into v respectively. Let Y_e represent the symbol transmitted on edge e. Each node operates in the following manner.

$$Y_e = \sum_{e' \in \Gamma^i(v)} \alpha \times Y_{e'} \text{ for } e \in \Gamma^o(v) \quad (5)$$

i.e. each node scales the symbol on each input edge by α (note that α is the same for every node) and the forwards the sum of the scaled inputs on all output edges. We shall see that the setting $\alpha = 1$ will ensure that each terminal recovers $\sum_{i=1}^n X_i$. To see this we examine the transfer matrix from the inputs $[X_1 \ \ldots \ X_n]_{1 \times n}$ to the output $Z_{T_j \rightarrow T'_j}$ denoted M_j which is of dimension $n \times 1$ i.e. $Z_{T_j \rightarrow T'_j} = [X_1 \ \ldots \ X_n] M_j$. Note that the i^{th} entry of M_j corresponds to the sum of the gains from all possible paths from S'_i to T'_j. The construction of G^* ensures that there is exactly one such path. Therefore the i^{th} entry of M_j will be a non-zero monomial in α for all $i = 1, \ldots, n$. Now setting $\alpha = 1$ will ensure that all the monomials evaluate to 1 i.e. $M_j = [1 \ \cdots \ 1]$, which implies that $Z_{T_j \rightarrow T'_j} = \sum_{i=1}^n X_i$. ■

As in the previous section it is clear that if any of the conditions in the statement of Theorem 2 are violated then either terminal T_1 or T_2 will be unable to find $\sum_{i=1}^n X_i$. For example if max-flow$(X_j - T_1) = 0$ then the received signal at T_1 cannot depend on X_j. Thus, T_1 cannot compute any function that depends on X_j.

V. Conclusion

We considered the problem of finding the rate region for the problem of communicating the modulo-2 sum of a set of independent unit rate sources to a set of terminals in the case when the underlying network can be modeled as a directed acyclic graph with unit capacity edges. The rate region has been presented for the cases when there are (a) two sources and n terminals, and (b) n sources and two terminals. Rate regions for arbitrary number of sources and terminals over general network topologies possibly containing cycles are currently under investigation.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Trans. on Info. Th.*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] S.-Y. Li, R. W. Yeung, and N. Cai, “Linear Network Coding,” *IEEE Trans. on Info. Th.*, vol. 49, no. 2, pp. 371–381, 2003.

[3] R. Koetter and M. Médard, “Beyond Routing: An Algebraic Approach to Network Coding,” in *IEEE Infocom*, 2002.

[4] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. on Info. Th.*, vol. 19, pp. 471–480, Jul. 1973.

[5] T. Ho, M. Médard, M. Effros, and R. Koetter, “Network Coding for Correlated Sources,” in *CISS*, 2004.

[6] A. Ramamoorthy, K. Jain, P. A. Chou, and M. Effros, “Separating Distributed Source Coding from Network Coding,” *IEEE Trans. on Info. Th.*, vol. 52, pp. 2785–2795, June 2006.

[7] Y. Wu, V. Stanković, Z. Xiong, and S. Y. Kung, “On practical design for joint distributed source coding and network coding,” in *Proceedings of the First Workshop on Network Coding, Theory and Applications, Riva del Garda, Italy*, 2005.