# Communicating the sum of sources in a 3-sources/3-terminals network; revisited

Michael Langberg
Computer Science Division
Open University of Israel
Email: mikel@openu.ac.il

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

***Abstract*—We consider the problem of multicasting sums over directed acyclic networks with unit capacity edges. A set of source nodes $s_i$ observe independent unit-entropy source processes $X_i$ and want to communicate $\sum X_i$ to a set of terminals $t_j$. Previous work on this problem has established necessary and sufficient conditions on the $s_i - t_j$ connectivity in the case when there are two sources or two terminals (Ramamoorthy '08), and in the case of three sources and three terminals (Langberg-Ramamoorthy '09). In particular the latter result establishes that each terminal can recover the sum if there are two edge disjoint paths between each $s_i - t_j$ pair. In this work, we provide a new and significantly simpler proof of this result, and introduce techniques that may be of independent interest in other network coding problems.**

## I. Introduction

We consider the problem of communicating sums over networks. 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 over a finite field, 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.

The rate region for this problem was characterized by Ramamoorthy in [1] for the case of directed acyclic networks (DAGs) with unit capacity edges and independent, unit entropy sources in which the network has at most two sources or two terminals. In this case a single path between each source terminal pair is both necessary and sufficient. Subsequently, the work of Langberg & Ramamoorthy [2], showed that the characterization of [1] does not hold in the case of networks with three sources and three terminals (see also [3]) and proposed an achievable region. Reference [2], shows that as long as each source terminal pair is connected by *two* edge disjoint paths, the terminals can recover the sum.

The main aim of this paper is to provide a significantly simpler proof of the result of [2]. Our proof technique is novel and may be of independent interest in other network coding problems. To summarize, the main result of this paper is a new proof for the following theorem:

*Theorem 1:* Let $G = (V, E)$ be a directed acyclic network with unit capacity edges and three sources $s_1, s_2, s_3$ containing independent unit-entropy source processes $X_1, X_2, X_3$ and three terminals $t_1, t_2, t_3$. If there exist two edge disjoint paths between each source/terminal pair, then there exists a linear network coding scheme in which the sum $X_1 + X_2 + X_3$ is obtained at each terminal $t_j$. Moreover, such a network code can be found efficiently.

In the above theorem we assume that the source process $X_i$ emits symbols from a finite field and the sum is also computed over the finite field.

Authors are in alphabetical order. The work of M. Langberg was supported in part by ISF grant 480/08. The work of A. Ramamoorthy was supported in part by NSF grant CNS-0721453.

## II. Proof of Theorem 1

In the interest of a self contained presentation, we repeat some of the material from [2] that is essential in setting up the basic definitions required for the rest of the discussion. Nevertheless, for a detailed model for linear network coding, please refer to [2].

Our proof for determining the desired network code has three steps. In the first step, we turn our graph $G$ into a graph $\hat{G} = (\hat{V}, \hat{E})$ in which each internal node $v \in \hat{V}$ is of total degree at most three. We refer to these graphs as *structured graphs*. This is outlined in [2] and explained in detail in [4]. It can be shown that proving Theorem 1 on structured graphs is equivalent to providing a proof for general graphs $G$ (see [2], [4]). For notational reasons, from this point on in the discussion we will assume that our input graph $G$ is structured — which is now clear to be w.l.o.g.

In the second step of our proof, we give edges and vertices in the graph $G$ certain labels depending on the combinatorial structure of $G$. This step induces a decomposition of the graph $G$ (both the vertex set and the edge set) into certain *class* sets that play a major role in our analysis. The decomposition of $G$ is given in detail in Section II-A.

In the third and final step of our proof, using the labeling above we present a case analysis for the proof of Theorem 1. Namely, based on the terminology set in Section II-A, we identify several scenarios, and prove Theorem 1 assuming that they hold. It will be evident that our proof also results in an efficient construction of the desired network code for $G$.

### A. The decomposition

In this section we present our structural decomposition of $G = (V, E)$. We assume throughout that $G$ is directed and acyclic, that it has three sources $s_1, s_2, s_3$, three terminals $t_1, t_2, t_3$ and that any internal vertex in $V$ (namely, any vertex which is neither a source or a sink) has total degree at most three. Moreover, we assume that $G$ satisfies the connectivity requirements specified in Theorem 1.

We start by labeling the vertices of $G$. A vertex $v \in V$ is labeled by a pair $(c_s, c_t)$ specifying how many sources (terminals) it is *connected* to. Specifically, $c_s(v)$ equals the number of sources $s_i$ for which there exists a path connecting $s_i$ and $v$ in $G$. Similarly, $c_t(v)$ equals the number of terminals $t_j$ for which there exists a path connecting $v$ and $t_j$ in $G$. For example, any source is labeled by the pair $(1,3)$, and any terminal by the pair $(3,1)$. An internal vertex $v$ labeled $(\cdot,1)$ is connected to a single terminal only. This implies that any information leaving $v$ will reach at most a single terminal. Such vertices $v$ play an important role in the definitions to come. This concludes the labeling of $V$.

An edge $e = (u,v)$ for which $v$ is labeled $(\cdot,1)$ will be referred to as a *terminal* edge. Namely, any information flowing on $e$ is constrained to reach at most a single terminal. If this terminal is $t_j$ then we will say that $e$ is a $t_j$-edge. Clearly, the set of $t_1$-edges is disjoint from the set of $t_2$-edges (and similarly for any pair of terminals). An edge which is not a terminal edge will be referred to as a *remaining* edge or an $r$-edge for short.

We now state some structural properties of the edge sets we have defined. First of all, there exists an ordering of edges in $E$ in which any $r$-edge comes before any terminal edge, and in addition there is no path from a terminal edge to an $r$-edge. This is obtained by an appropriate topological order in $G$. Moreover, for any terminal $t_j$, the set of $t_j$-edges form a connected subgraph of $G$ rooted at $t_j$. To see this note that by definition each $t_j$-edge $e$ is connected to $t_j$ and all the edges on a path between $e$ and $t_j$ are $t_j$-edges. Finally, the head of an $r$-edge is either of type $(\cdot,2)$ or $(\cdot,3)$ (as otherwise it would be a terminal edge).

For each terminal $t_j$ we now define a set of vertices referred to as the leaf set $L_j$ of $t_j$. This definition shall play an important role in our discussions.

*Definition 1: Leaf set of a terminal.* Let $P = (v_1, v_2, \ldots, v_\ell)$ be a path from $s_i$ to $t_j$ (here $s_i = v_1$ and $t_j = v_\ell$). Consider the intersection of $P$ with the set of $t_j$-edges, This intersection consists of a subpath $P'$, $(v_P, \ldots, v_\ell = t_j)$ of $P$ for which the label of $v_P$ is either $(\cdot,2)$ or $(\cdot,3)$, and the label of any other vertex in $P'$ is $(\cdot,1)$. We refer to $v_P$ as the leaf of $t_j$ corresponding to path $P$, and the set of all leaves of $t_j$ as the leaf set $L_j$.

We remark that (a) the leaf set of $t_j$ is the set of nodes of in-degree 0 in the subgraph consisting of $t_j$-edges and (b) a source node can be a leaf node for a given terminal.

*B. Case analysis*

We now classify networks based on the node labeling procedure introduced above. For each class of networks, we argue that (given the requirement stated in Theorem 1) a network code can be found (efficiently) that allows the recovery of $\sum_{i=1}^{3} X_i$ at the terminals. The proofs of cases 0, 1 and 2 below can be found in [2] and are skipped. In [2] and [5] we also present an elaborate proof for the final and most complicated case 3. The contribution of the current paper is in a significantly simpler analysis of case 3 which involves a refined labeling of the vertex set $V$. Our new refined labeling and analysis techniques may be of independent interest and will hopefully yield a better understanding of additional problems as well (such as the characterization of the multiple-unicast capacity in 3-source/3-terminal networks).

- *Case 0.* There exists a node of type $(3,3)$ in $G$.
- *Case 1.* There exists a node of type $(2,3)$ in $G$.
- *Case 2.* There exists a node of type $(3,2)$ in $G$.
- *Case 3.* There do not exist nodes of type $(3,3)$, $(2,3)$ and $(3,2)$ in $G$.

## III. Analysis of Case 3

We now prove Theorem 1 under the assumption that $G$ has no nodes of type $(3,3)$, $(2,3)$ and $(3,2)$. Note that the node labeling procedure presented above assigns a label $(c_s(v), c_t(v))$ to a node $v$ where $c_s(v)$ ($c_t(v)$) is the number of sources (terminals) that $v$ is connected to. This labeling ignores the actual identity of the sources and terminals that have connections to $v$. It turns out that the labeling is sufficient to handle cases 0, 1 and 2 (see [2]). However, we need to use an additional, somewhat finer notion of node connectivity when we want to analyze case 3. We emphasize that throughout this section, we still operate under the assumption that the reduction outlined in Section II has been performed and that each node has a total degree at most three.

Towards this end, for case 3 (i.e., in a graph $G$ without $(3,3)$, $(2,3)$ and $(3,2)$ nodes) we introduce the notion of the color of a node. For each $(2,2)$ node in $G$, the color of the node is defined as the 4-tuple of sources and terminals it is connected to, e.g., if $v$ is connected to sources $s_1$ and $s_2$ and terminals $t_1$ and $t_2$, then its color, denoted col($v$) is $(s_1, s_2, t_1, t_2)$. We shall also say that the source color of $v$ is $(s_1, s_2)$ and the terminal color of $v$ is $(t_1, t_2)$. The source and terminal colors are sometimes referred to as source and terminal labels. The following claim is immediate.

*Claim 1:* If there is a $(2,2)$ node $v$ in $G$ of color col($v$), then each terminal in the terminal color of $v$ has at least one leaf with color col($v$). For example, if col($v$) = $(s_1, s_2, t_1, t_2)$, then both $t_1$ and $t_2$ have leaves with color $(s_1, s_2, t_1, t_2)$.

*Proof:* W.l.o.g, let col($v$) = $(s_1, s_2, t_1, t_2)$. This implies that there exists a path $P$ between $v$ and $t_1$. Let $\ell$ be a leaf of $t_1$ on $P$. Recall that $\ell$ is defined as the last node on $P$ with terminal label at least 2, namely $c_t(\ell) \geq 2$. As $\ell$ is down stream of $v$ it holds that $c_t(v) \geq c_t(\ell)$ and that the terminal color of $v$ includes that of $\ell$. Thus we conclude that $c_t(\ell)$ is exactly 2 and no larger as otherwise $c_t(v)$ would also be greater than 2 contradicting our assumptions in the claim. This implies that the terminal color of $\ell$ is exactly $(t_1, t_2)$.

As $\ell$ is downstream of $v$ it also holds that $c_s(\ell) \geq c_s(v) = 2$ and that the source color of $\ell$ includes that of $v$. Thus, it holds that $c_s(\ell)$ is exactly 2, otherwise $\ell$ would be a $(3,2)$ node (contradicting our assumption for case 3). This implies that the source color of $\ell$ is $(s_1, s_2)$. Therefore, $t_1$ has a leaf of color $(s_1, s_2, t_1, t_2)$. A similar argument holds for $t_2$. ■

The notion of a color is useful for the set of graphs under case 3, since we can show that there can never be an edge between nodes of different colors. We exploit this property extensively below.

*Lemma 1:* Consider a graph $G$, with sources, $s_i, i = 1, \ldots, 3$, and terminals $t_j, j = 1, \ldots 3$, such that it does not have any $(3,3)$, $(2,3)$ or $(3,2)$ nodes. There does not exist an edge between $(2,2)$ nodes of different color in $G$.

*Proof:* Assume otherwise and consider two $(2,2)$ nodes $v_1$ and $v_2$ such that $\text{col}(v_1) \neq \text{col}(v_2)$, for which there is an edge $(v_1, v_2)$ in $G$. Note that if the source colors of $\text{col}(v_1)$ and $\text{col}(v_2)$ are different, then $v_2$ has to be a $(3,2)$ node, which is a contradiction. Likewise, if the terminal colors of $\text{col}(v_1)$ and $\text{col}(v_2)$ are different, then $v_1$ has to be a $(2,3)$ node, which is also a contradiction. ■

Lemma 1 implies that we are free to assign any coding coefficients on a subgraph induced by nodes of one color, without having to worry about the effect of this on another subgraph induced by nodes of a different color (simply because there is no such effect).

The basic idea of our proof is the following. We divide the set of graphs under case 3, into various classes, depending on the number of colors that exist in the graph. It turns out that as long as the number of colors in the graph is not 2, i.e., either 0,1 or 3 and higher, then there is a simple argument which shows that each terminal can be satisfied. The argument in the case of two colors is a bit more involved and is developed separately. It can be shown that our counter-example in [2] is a case where there are two colors. Note however, that in our counter-example there are certain $s_i - t_j$ pairs that have only one path between them. We now proceed to develop these arguments formally.

*Claim 2:* Consider the subgraph induced by a certain color, w.l.o.g. $(s_1, s_2, t_1, t_2)$ in $G$, denoted by $G_{(s_1,s_2,t_1,t_2)}$. There exists an assignment of encoding vectors over $G_{(s_1,s_2,t_1,t_2)}$, such that any (unit entropy) function of the source processes $X_1$ and $X_2$ can be multicasted to all nodes in $G_{(s_1,s_2,t_1,t_2)}$. Moreover, such encoding vector assignments can be done independently over subgraphs of different colors.

*Proof:* Note that we are working with directed acyclic graphs. Thus, there is a node $v^*$ in $G_{(s_1,s_2,t_1,t_2)}$, such that it has no incoming edges in $G_{(s_1,s_2,t_1,t_2)}$. There are paths from both $s_1$ and $s_2$ to $v^*$. Note that the path from $s_1$ to $v^*$ has no intersection with any path from $s_2$ or $s_3$. To see this, suppose that there was such an intersection at node $v'$. If there is a path from $s_3$ to $v'$, then $v^*$ is a $(3,2)$ node (which contradicts the assumption that $v^*$ is a $(2,2)$ node). If there is a path from $s_2$ to $v'$, then $v'$ and the remaining vertices connecting $v'$ to $v^*$ on the path from $s_1$ to $v^*$ have color $(s_1, s_2, t_1, t_2)$. Contradicting the fact that $v^*$ has no incoming edges in $G_{(s_1,s_2,t_1,t_2)}$. Likewise, we see that the path from $s_2$ to $v^*$ has no intersection with a path from $s_1$ or $s_3$.

Therefore, the path from $s_1$ to $v^*$ carries $X_1$ in the clear, and likewise for the path from $s_2$ to $v^*$. Thus, $v^*$ can obtain both $X_1$ and $X_2$ and can compute any (unit entropy) function of them. Moreover, $v^*$ can transmit this function to all nodes of $G_{(s_1,s_2,t_1,t_2)}$ downstream of $v^*$. As the argument above can be repeated for any node $v^*$ of in-degree 0 in $G_{(s_1,s_2,t_1,t_2)}$ it follows that all nodes of $G_{(s_1,s_2,t_1,t_2)}$ can obtain the desired function of $X_1$ and $X_2$.

Finally, we note that the assignments over subgraphs of different colors can be done independently, since there does not exist any edge between nodes of different colors (from Lemma 1). ■

In what follows, a *greedy* encoding at node $v$ either takes the sum of incoming symbols or just forwards one of the incoming symbols depending on which action yields the "largest support". For example, if the two input edges contain $X_1$ and $X_2$, then the outgoing edge will carry $X_1 + X_2$, if they carry $X_1$ and $X_1 + X_2$, the outgoing edge will still carry $X_1 + X_2$, and if they carry $X_3$ and $X_1 + X_2$ then the outgoing edge will carry $X_1 + X_2 + X_3$. If the incoming information of $v$ is $X_1 + X_2$ on one edge and $X_2 + X_3$ on another (or more generally, the support of the incoming edges is not disjoint or included), then greedy encoding will not be used.

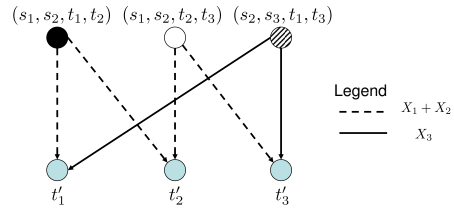


Fig. 1. A possible instance of $G_{aux}$ when the degree sequence of the terminals is $(2,2,2)$. The encoding specified in the legend denotes the encoding to be propagated downstream to the leaf nodes.

*Lemma 2:* Consider a graph $G$, with sources, $s_i, i = 1, \ldots, 3$, and terminals $t_j, j = 1, \ldots 3$, such that (a) it does not have any $(3,3), (2,3)$ or $(3,2)$ nodes, and (b) there exists at least one $s_i - t_j$ path for all $i$ and $j$. Consider the set of all $(2,2)$ nodes in $G$ and their corresponding colors. If there exists no colors, exactly one color or at least three distinct colors in $G$, then there exists a set of coding vectors such that each terminal can recover $\sum_{i=1}^{3} X_i$.

*Proof:* Note that all leaves in $G$ are of type $(1,2), (1,3)$ or $(2,2)$. This implies that any terminal $t_j$ that does not have a $(2,2)$ leaf with source color including $s_i$, must have a leaf at which $X_i$ is received in the clear. The above follows directly by the connectivity assumption (b) stated in the Lemma.

(0) *Case 0.* There are no colors in $G$.
This implies that there are no $(2,2)$ nodes in $G$ and thus all terminals $t_j$ have distinct leaves holding $X_1$, $X_2$, and $X_3$ respectively. This suffices to design a simple greedy code on the paths from those leaves to $t_j$ which enables $t_j$ to recover the sum $X_1 + X_2 + X_3$.

(i) *Case 1.* There is only one color in $G$.
In this case perform greedy encoding on the $r$-edges. We show that each terminal can recover $\sum_{i=1}^{3} X_i$ from the content of its leaves. W.l.o.g, suppose that the color is $(s_1, s_2, t_1, t_2)$. Using Claim 1, this means that both $t_1$ and $t_2$ have leaves of this color. The greedy encoding implies that $t_1$ and $t_2$ can obtain $X_1 + X_2$ from the corresponding leaves. Moreover, both $t_1$ and $t_2$ have a leaf containing a singleton $X_3$, because of the connectivity requirements. Therefore, they can compute $\sum_{i=1}^{3} X_i$. The terminal $t_3$ has only singleton leaves, such that there exists at least one $X_1, X_2$ and $X_3$ leaf. Thus it can compute their sum.

(ii) *Case 2.* There exist exactly three distinct colors in $G$.
It is useful to introduce an auxiliary bipartite graph that denotes the existence of the colors at the leaves of the different terminals. This bipartite graph denoted $G_{aux}$ is constructed as follows. There are three nodes $t'_i, i = 1, \ldots, 3$ that denote the terminals on one side and three nodes $c'_i, i = 1, \ldots, 3$ that denote the colors on the other side. If the color $c'_i$ has $t_j$ in its support, then there is an edge between $c'_i$ and $t'_j$, i.e., $t_j$

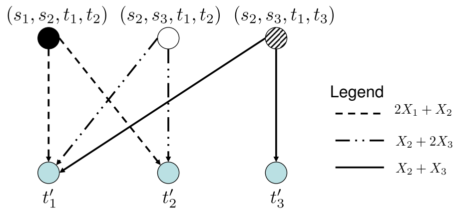


Fig. 2. A possible instance of $G_{aux}$ when the degree sequence of the terminals is $(3,2,1)$. The encoding specified in the legend denotes the encoding to be propagated downstream to the leaf nodes.

has a leaf of color $c'_i$. The following properties of $G_{aux}$ are immediate.

(i) Each $c'_i$ has degree-2. (ii) Each $t'_i$ has degree at most 3. (iii) There are no multiple edges in $G_{aux}$.

Note that there are exactly three possible source colors ($(s_1,s_2),(s_2,s_3)$ and $(s_3,s_1)$) and three possible terminal colors ($(t_1,t_2),(t_2,t_3)$ and $(t_3,t_1)$). We now perform a case analysis depending upon the degree sequence of nodes $t'_j, j=1,\dots,3$ in $G_{aux}$. The degree sequence is specified by a 3-tuple, where the sum of the entries has to be six.

a) The degree sequence is a permutation of $(0,3,3)$.

This only happens if the terminal label of all colors, $c'_i, i=1,\dots,3$ is the same and in turn implies that the source label of each color is distinct, i.e., the source colors include $(s_1,s_2),(s_2,s_3)$ and $(s_1,s_3)$. In this case, greedy encoding works for the two terminals in the color support. This is because each terminal will obtain $X_1+X_2$, $X_2+X_3$ and $X_1+X_3$ at its leaves (using Lemma 1), from which the terminal can compute $2\sum_{i=1}^{3} X_i$ (here we assume that the field characteristic is greater than two). The remaining terminal is not connected to any (2,2) leaf, so that all its leaves contain singleton values, from which it can compute $\sum_{i=1}^{3} X_i$.

b) The degree sequence is $(2,2,2)$.

This only happens if all the terminal labels of the colors are distinct, i.e., the terminal labels are $(t_1,t_2)$, $(t_2,t_3)$ and $(t_1,t_3)$. Now consider the possibilities for the source labels.

If there is only one source label, then greedy encoding ensures that the sum of exactly two of the sources reaches each terminal. The connectivity condition guarantees that the remaining source is available as a singleton at a leaf of each terminal. Therefore we are done.

If there are exactly two distinct source colors, then we argue as follows. On the subgraphs induced by the colors with the same source label, perform greedy encoding. On the remaining subgraph, propagate the remaining useful source. We illustrate this with an example that is w.l.o.g. Suppose that the colors are $(s_1,s_2,t_1,t_2)$, $(s_1,s_2,t_2,t_3)$ and $(s_2,s_3,t_1,t_3)$. We perform greedy encoding on the subgraphs of the first two colors, and only propagate $X_3$ on the subgraph of the third color. As shown in Figure 1, this means that terminals $t_1$ and $t_3$ are satisfied. Note that the connectivity condition dictates that $t_2$ has to have a leaf that has a singleton $X_3$, therefore it is satisfied as well.

Finally, suppose that there are three distinct source colors. In this case we use the encoding specified in Table I on the subgraphs of each source color. It is clear on inspection that $\sum_{i=1}^{3} X_i$ can be recovered from any two of the received values (as from any two of the linear combinations stated, one can deduce the sum $X_1+X_2+X_3$). Here also we assume that the field characteristic is greater than two.

TABLE I
ENCODING ON SUBGRAPHS OF DIFFERENT SOURCE COLORS. RECOVERY OF $\sum_{i=1}^{3} X_i$ IS POSSIBLE FROM ANY TWO OF THE RECEIVED VALUES, USING ADDITIONS OR SUBTRACTIONS.

| Source color | Encoding |
|---|---|
| $(s_1,s_2)$ | $2X_1+X_2$ |
| $(s_2,s_3)$ | $X_2+2X_3$ |
| $(s_1,s_3)$ | $X_1-X_3$ |

c) The degree sequence is a permutation of $(1,2,3)$.

In this case, the degree sequence dictates that there have to be two terminals that share two colors. This implies that the source label of those colors has to be different. For the subgraphs induced by these colors, we use the encoding proposed in Table I. For the subgraph induced by the remaining color, we perform greedy encoding. For example, suppose that the colors are $(s_1,s_2,t_1,t_2)$, $(s_2,s_3,t_1,t_2)$ and $(s_2,s_3,t_1,t_3)$. As shown in Figure 2, $t_1$ and $t_2$ are clearly satisfied (even without using the information from color $(s_2,s_3,t_1,t_3)$). Terminal $t_3$ has to have a singleton leaf containing $X_1$ by the connectivity condition and is therefore satisfied.

Together, these arguments establish that in the case when there are three colors, all terminals can be satisfied.

(iii) *Case 3.* There exist more than three distinct colors in $G$. Note that if there are at least four colors in $G$, then (a) there are two colors with the same terminal label, since there are exactly three possible terminal labels, and (b) for the colors with the same terminal labels, the source labels necessarily have to be different. Our strategy is as follows. For the terminals that share two colors, use the encoding proposed in Table I. If the remaining terminal has access to only one source color, then use greedy encoding and note that this terminal has to have a singleton leaf. If it has access to at least two source colors, simply use the encoding in Table I for it as well.

We thus conclude our proof. ■

It remains to develop the argument in the case when there are exactly two distinct colors in $G$. For this we need to explicitly use the fact that there are two edge-disjoint paths between each $s_i - t_j$ pair.

*Lemma 3:* Consider a graph $G$, with sources, $s_i, i=1,\dots,3$, and terminals $t_j, j=1,\dots 3$, such that (a) it does not have any $(3,3),(2,3)$ or $(3,2)$ nodes, and (b) there exist at least two $s_i - t_j$ paths for all $i$ and $j$. Consider the set of all $(2,2)$ nodes in $G$ and their corresponding colors. If there exist exactly two distinct colors in $G$, then there exists a set of coding vectors such that each terminal can recover $\sum_{i=1}^{3} X_i$.

*Proof:* As in the proof of Lemma 2, we argue based on the content of the leaves of the terminals. Suppose that the auxiliary bipartite graph $G_{aux}$ is formed. If both the colors have the same terminal label (see Figure 3 for an example), then it is clear that the encoding in Table I on the subgraphs induced by the colors suffices for the corresponding terminals. The third terminal has singleton leaves corresponding to each source and can compute $\sum_{i=1}^{3} X_i$.

Another possibility is that the terminal labels of the colors are different, but the source labels are the same. This case can be handled by greedy encoding on the colors.

The situation is more complicated when the terminal and

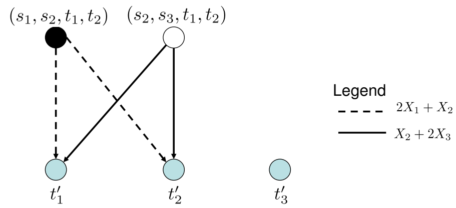


Fig. 3. An instance of $G_{aux}$ when there exist exactly two distinct colors under case 3, such that the terminal labels of the colors are the same.

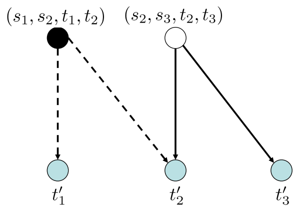


Fig. 4. An instance of $G_{aux}$ when there exist exactly two distinct colors under case 3, such that both the source labels and the terminal labels of the colors are different.

source labels of the colors are different, see, e.g., Figure 4. In the case depicted, greedy encoding does not work since it satisfies $t_1$ and $t_3$ but not $t_2$. W.l.o.g., we assume that the colors are $(s_1, s_2, t_1, t_2)$ and $(s_2, s_3, t_2, t_3)$. Now, we know that there exist two vertex-disjoint paths between $s_1$ (a similar argument can be made for $s_3$) and $t_2$. Each of these paths has a leaf for $t_2$. If one of the leaves contains a singleton $X_1$, then performing greedy encoding on the two colors works since $t_2$ obtains $X_1 + X_2$, $X_1$ and $X_2 + X_3$ on its leaves and the other terminals will obtain singleton leaves that satisfy their demand. Likewise, if there is a singleton leaf containing $X_3$ on the vertex disjoint paths from $s_3$ to $t_2$, then greedy encoding works.

Thus, the leaves of $t_2$ must be of type $(2, 2)$. This implies that there are at least four distinct leaves of $t_2$ of type $(2, 2)$, two of color $(s_1, s_2, t_1, t_2)$ and two of color $(s_2, s_3, t_2, t_3)$. We now conclude our proof by the following claims.

Consider the subgraph induced by nodes colored by one of the colors above, w.l.o.g. $(s_1, s_2, t_1, t_2)$, in $G$ together with the $(1, \cdot)$ nodes in $G$. Denote this subgraph by $G'$. Consider a random linear network code on the nodes of $G'$ (namely, each node outputs a random linear combination of its incoming information over the underlying finite field). We show, with high probability (given the field size is large enough), that such a code allows both $t_1$ and $t_2$ to receive two linearly independent combinations of $X_1$ and $X_2$ at their leaves. An analogous argument also holds for $t_2$ and $t_3$ when considering the color $(s_2, s_3, t_2, t_3)$ and the information $X_2$ and $X_3$. This suffices to conclude our assertion. In what follows, we denote the size of $V$ by $n$ and the underlying field size by $q$.

*Claim 3:* Let $u$ be any leaf in $G'$. Let $U = \alpha X_1 + \beta X_2$ be the incoming information of $u$. With probability $(1 - 2/q)^n$ both $\alpha$ and $\beta$ are not zero.

*Proof:* The proof is standard and omitted due to space limitations. We use the techniques presented in [6]. ■

Consider the terminal $t_2$ and its two edge disjoint paths from $s_1$ denoted $P_1$ and $P_2$. Let $u_1$ and $u_2$ be the corresponding leaves on paths $P_1$ and $P_2$. As the leaves of $t_2$ are of type $(2, 2)$ and as both $u_1$ and $u_2$ are connected to $s_1$ it holds that both $u_1$ and $u_2$ are of color $(s_1, s_2, t_1, t_2)$ and in $G'$. The following claim shows that with high probability (given $q$ large enough) $t_2$ will receive two linearly independent combinations of $X_1$ and $X_2$ at $u_1$ and $u_2$. The proof is omitted due to space limitations (it follows the line of analysis presented in [6]).

*Claim 4:* Let $U_1 = \alpha_1 X_1 + \beta_1 X_2$ be the incoming information of $u_1$, and $U_2 = \alpha_2 X_1 + \beta_2 X_2$ the incoming information of $u_2$. With probability $(1 - 2/q)^n$ the vectors $\{(\alpha_i, \beta_i)\}_{i=1,2}$ are independent.

Now, consider the terminal $t_1$ and its two edge disjoint paths from $s_1$ denoted $P_1$ and $P_2$. Let $u_1$ and $u_2$ be the corresponding leaves on paths $P_1$ and $P_2$ (to simplify notation we use the same notation as previously used for $t_2$). Here, we consider two cases, if both $u_1$ and $u_2$ are $(2, 2)$ nodes, then by Claim 4 we are done (with high probability). Namely, with high probability (given $q$ large enough) $t_1$ will receive two linearly independent combinations of $X_1$ and $X_2$ at $u_1$ and $u_2$. Otherwise, $t_1$ has at least one leaf with $X_1$ in the clear. Denote this leaf as $v_1$. Notice that $t_1$ must have at least a single $(2, 2)$ leaf (by Claim 1), denote this leaf by $v_2$. Finally, by Claim 3 it holds that with high probability the information present at $v_1$ and at $v_2$ is independent.

To conclude, notice that the discussion above (when applied symmetrically for $t_2$, $t_3$, and the color $(s_2, s_3, t_2, t_3)$) implies that all terminals are able to obtain the desired sum $X_1 + X_2 + X_3$ (by an appropriate setting of the encoding functions on their $(\cdot, 1)$ edges). ■

## IV. Conclusions

In this work we have addressed the network arithmetic problem in the scenario in which the network has three sources and three terminals. We have presented a new and significantly simpler proof for Theorem 1 based on a refined labeling scheme which decomposes the given graph $G$ into *independent* components.

Several questions remain open. Primarily, is the 2-connectivity condition (between $s_i/t_j$ pairs) *tight* or can other combinatorial connectivity requirements characterize the capacity of the network arithmetic problem for the $3s/3t$ case. Secondly, it is natural to ask what happens with more than 3 sources and terminals. More specifically, our proof for $3s/3t$ is strongly based on our notion of labeling and coloring. These notions extend naturally to $k$ sources and $k$ terminals, however, our line of proof becomes much more complicated for $k > 3$. The question whether there exists a unified line of analysis for all $k$ (or even $k = 4$) is left open in this work.

## References


[1] A. Ramamoorthy, "Communicating the sum of sources over a network," in *IEEE Intl. Symposium on Info. Th.*, 2008, pp. 1646–1650.

[2] M. Langberg and A. Ramamoorthy, "Communicating the sum of sources in a 3-sources/3-terminals network," in *IEEE Intl. Symposium on Info. Th.*, 2009, pp. 2121–2125.

[3] B. K. Rai, B. K. Dey, and A. Karandikar, "Some results on communicating the sum of sources over a network," in *NetCod*, 2009.

[4] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *IEEE Trans. on Info. Th.*, vol. 52, no. 6, pp. 2386–2397, 2006.

[5] M. Langberg and A. Ramamoorthy, "Communicating the sum of sources in a 3-sources/3-terminals network," *Manuscript, 2009, available at* `http://www.openu.ac.il/home/mikel/ISIT09/ISIT09.pdf`.

[6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. on Info. Th.*, vol. 52, no. 10, pp. 4413–4430, 2006.