

EGERVÁRY RESEARCH GROUP  
ON COMBINATORIAL OPTIMIZATION



TECHNICAL REPORTS

TR-2011-01. Published by the Egerváry Research Group, Pázmány P. sétány 1/C,  
H-1117, Budapest, Hungary. Web site: [www.cs.elte.hu/egres](http://www.cs.elte.hu/egres). ISSN 1587-4451.

---

**Multi-Layered Video Broadcast using  
Network Coding and a Distributed  
Connectivity Algorithm**

Zoltán Király and Erika Renáta Kovács

---

March 2011

# Multi-Layered Video Broadcast using Network Coding and a Distributed Connectivity Algorithm

Zoltán Király and Erika Renáta Kovács

## Abstract

Multi-layered video streaming considers different quality requirements of the receivers. Network coding has been shown to be a useful tool to increase throughput of multi-layered service compared to simple multicasting. Kim et al. [3] gave a simple effective algorithm using network coding. We generalize their approach and give an algorithm that solves the problem for two layers optimally for certain natural objective functions and prove NP-hardness of the problem for some other objectives, as well as for more than two layers. We also give a heuristic for three layers.

## 1 Introduction

The increasing demand for on-line video downloads, internet TV broadcasting, teleconferencing and similar applications, and, on the other hand, different available download speeds and/or output resolutions of receivers motivate a service, where different quality requirements are taken into account effectively. One solution to the problem is multi-layered video streaming.

A successful approach is to use multi-resolution codes (MRC), encoding data into a base layer and one or more refinement layers [4, 1]. Receivers can request cumulative layers, which are combined at the receivers to provide progressive refinement. The decoding of a higher layer always requires the correct reception of all lower layers including the base layer. The multi-layer multicast problem is to multicast as many valuable layers to as many receivers as possible.

It is possible to improve performance compared to simple routing by using network coding. In this short extended abstract we assume the reader is familiar with the concepts of network coding. A detailed description of the field can be found in [5]. The starting point of this research was the seminal paper of Kim et al. [3], where they introduced the problem of designing good time independent network codes for multi-resolution coding. Our complexity results explain why they were not able to prove optimality of their algorithms. Refining their methods we are able to prove optimality for the special case of two layers, where the problem is to transmit the first layer to every receiver, and transmit both layers to as many receivers as possible.

The rest of the paper is organized as follows: in Section 2 the main problem and some definitions are introduced. In Section 3 we prove NP-hardness for some special

cases of the problem. In Section 4, by the notion of arc limits, we generalize the approach of Kim et al. [3] and give an optimal algorithm for two layers. In Section 5 we give a sufficient condition for receiver demands that can be fulfilled simultaneously and we present a distributed algorithm to decide if an arc limit satisfies this condition. Finally, we give a heuristic for three layers, which is also based on our distributed algorithm.

## 2 Problem Definition

Actual packets of the layers of the video stream will correspond to messages  $\mathbf{M} = (M_1, M_2, \dots, M_k)$ , represented by members of a finite field  $\mathbb{F}_q$  of size  $q$ , where  $k$  denotes the number of layers we are going to transmit. Let  $D = (V, A)$  be a directed acyclic graph with a single source node  $s$  and with unit capacity arcs. We will consider this graph fixed, except in Section 3, and we assume that every node of  $D$  is reachable from  $s$ . The task is to multicast  $\mathbf{M}$  from  $s$ . The idea of network coding is to transmit linear combinations of messages in  $\mathbf{M}$  on the arcs. Such a linear combination can be represented by the vector of the coefficients  $\mathbf{c} = (c_1, \dots, c_k)$ ,  $c_i \in \mathbb{F}_q$ . Let  $\mathbb{F}_q^k$  denote the  $k$ -dimensional vector space over  $\mathbb{F}_q$ , and let  $\mathbf{e}_i$  denote the  $i$ th unit vector. For a set  $S \subseteq \mathbb{F}_q^k$  of vectors, let  $\langle S \rangle$  denote the linear subspace spanned by  $S$ .

**Definition 2.1.** A **linear network code** of  $k$  messages on an acyclic graph  $D$  over a finite field  $\mathbb{F}_q$  is a mapping  $\mathbf{c} : A \rightarrow \mathbb{F}_q^k$  which fulfills the **linear combination property**:  $\mathbf{c}(uv) \in \langle \{\mathbf{c}(wu) \mid wu \in A\} \rangle$  for all  $u \neq s$ . We will use the notation  $\langle \mathbf{c}, u \rangle = \langle \{\mathbf{c}(wu) \mid wu \in A\} \rangle$ . The function  $\mathbf{c}$  denotes the **coefficients** of messages on an arc, that is, on arc  $a$  the message sent is  $\mathbf{c}(a) \cdot \mathbf{M}$ .

For a linear network code  $\mathbf{c}$ , a node  $v$  **can decode** (or receives) the message  $M_i$ , if  $\mathbf{e}_i \in \langle \mathbf{c}, v \rangle$ .

Note that simple routing can be regarded as a special case, where for each arc  $uv$ ,  $\mathbf{c}(uv) = \mathbf{e}_i$  for some  $1 \leq i \leq k$ .

In multi-resolution coding a layer  $M_i$  is **valuable** for a node only if all the layers with higher importance can also be decoded at that node, i.e., for every  $j \leq i$  message  $M_j$  is decodable. A **request** of a node can be the first  $i$  layers, where  $0 \leq i \leq k$ .

**Definition 2.2.** Let  $T_i$  denote the set of nodes with request  $i$ . A **demand** is a sequence of disjoint subsets of  $V \setminus \{s\}$  denoted by  $\tau = (T_1, T_2, \dots, T_k)$ . For a node  $v \in V$ , let  $\lambda(s, v)$  denote the maximal number of arc-disjoint paths from  $s$  to  $v$ . A demand is **proper**, if  $\lambda(s, t_i) \geq i$  for all  $i$  and all  $t_i \in T_i$ . The set of **receiver nodes** is the union of these request sets, denoted by  $T = T_1 \cup T_2 \cup \dots \cup T_k$ .

**Definition 2.3.** A network code is **feasible** for demand  $\tau$  if for all  $i$ , every receiver node  $t_i \in T_i$  can decode  $M_j$  for all  $j \leq i$ .

**Definition 2.4.** The **height** of a linear code on an arc is the least important layer with nonzero coefficient on this arc. For example, the first unit vector has height one and so on,  $\mathbf{e}_i$  has height  $i$ , and vector  $(1, 0, 1, 0)$  has height 3. The height of a linear code  $\mathbf{c}$  is denoted by  $H_{\mathbf{c}} : A \rightarrow \mathbb{N}$ .

In [3] Kim et al. gave an algorithm for constructing a network code for which every receiver node can decode the first layer. In their approach each node is given a limit on the height of the linear code on entering arcs. Their algorithm ensures that the first layer can be decoded at each receiver with some probability, and some receivers may be able to decode more layers.

In this paper we give some algorithms that are also based on limiting the height of a linear code. In our algorithms limits may differ for arcs entering the same node.

### 3 Complexity Results

**Theorem 3.1.** *Given a network  $D$  and a demand  $\tau = (T_1, \emptyset, T_3)$ , it is NP-complete to decide, whether there exists a feasible network code for this demand.*

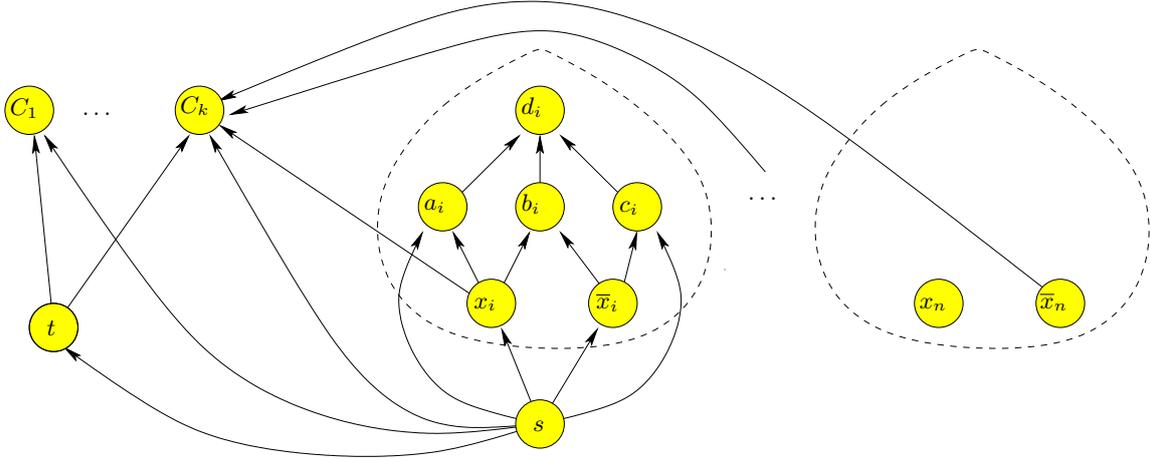
*Proof.* We reduce 3-SAT to this problem. Let  $S = (X, CL)$  be a 3-SAT instance, where  $X = \{x_1, \dots, x_n\}$  and  $CL = \{C_1, \dots, C_m\}$  denote the set of variables and clauses, respectively. For each variable  $x_i$  we add six nodes with eleven arcs (see Figure 1), so that  $a_i, b_i, c_i \in T_1$  and  $d_i \in T_3$ . Nodes  $x_i$  and  $\bar{x}_i$  correspond to literals. We add a special node  $t \in T_1$  and an arc  $st$ . For each clause  $C_j$  we add a node  $C_j$ , arcs  $sC_j$  and  $tC_j$  and arcs from every node corresponding to literals of  $C_j$ . Each  $C_j$  is in  $T_3$ . We prove that this network coding problem  $N$  has a feasible solution over some finite field if and only if  $S$  can be satisfied. Suppose  $N$  has a feasible network code  $\mathbf{c}$ . Let  $l_1, \dots, l_r$ ,  $1 \leq r \leq 3$  denote the literals in  $C_j$ . Since  $t \in T_1$ ,  $\mathbb{H}_c(st) = 1$  and for all  $C_j$   $\mathbb{H}_c(tC_j) = 1$ , and the arc  $sC_j$  can transmit any message from  $s$ , hence  $C_j$  can decode all three layers if and only if at least on one of the arcs  $l_iC_j$  the linear code has height greater than one.

**Claim 3.2.** *If  $N$  has a feasible network code  $\mathbf{c}$ , then for every variable  $x_i$ , the code  $\mathbf{c}$  has height one on at least one of the arcs  $sx_i$  and  $s\bar{x}_i$ .*

*Proof.* Let us assume indirectly that neither  $\mathbf{c}(sx_i)$ , nor  $\mathbf{c}(s\bar{x}_i)$  have height one. Since  $a_i, b_i, c_i$  must be able to decode the first layer,  $\mathbf{c}(sa_i) \in \langle \mathbf{e}_1, \mathbf{c}(sx_i) \rangle$ , and  $\mathbf{c}(sc_i) \in \langle \mathbf{e}_1, \mathbf{c}(s\bar{x}_i) \rangle$ , and  $\mathbf{e}_1 \in \langle \mathbf{c}(sx_i), \mathbf{c}(s\bar{x}_i) \rangle$ . Hence we have  $\dim \langle \mathbf{c}(sa_i), \mathbf{c}(sc_i), \mathbf{c}(sx_i), \mathbf{c}(s\bar{x}_i) \rangle = \dim \langle \mathbf{c}(sx_i), \mathbf{c}(s\bar{x}_i) \rangle \leq 2$ , that is, these four vectors cannot span a 3-dimensional space to transmit three layers to  $d_i$ .  $\square$

From the claim we can transform a solution of  $N$  into an assignment of  $S$  by assigning value 'true' to a literal  $l$  if the height of  $\mathbf{c}(sl)$  is at least two. Note that if for a variable  $x_i$  both  $sx_i$  and  $s\bar{x}_i$  have height one, we can choose the value of  $x_i$  arbitrarily to get a satisfying assignment.

Similarly we can get a feasible code  $\mathbf{c}$  for network  $N$  from a truth assignment of  $S$  over any field. The corresponding  $\mathbf{c}(e)$  vectors are the following. Let  $\mathbf{c}(st) = (1, 0, 0)$ ,  $\mathbf{c}(sC_j) = (1, 1, 1)$ , and for any node  $u$  with only one incoming arc  $wu$ , all outgoing arcs carry  $\mathbf{c}(wu)$ . If  $x_i$  is true, then  $\mathbf{c}(sa_i) = (0, 1, 0)$ ,  $\mathbf{c}(sx_i) = (1, 1, 0)$ ,  $\mathbf{c}(s\bar{x}_i) = (1, 0, 0)$ ,  $\mathbf{c}(sc_i) = (1, 1, 1)$ ,  $\mathbf{c}(a_id_i) = (0, 1, 0)$ ,  $\mathbf{c}(b_id_i) = (1, 0, 0)$ ,  $\mathbf{c}(c_id_i) = (1, 1, 1)$ , and the code can be constructed symmetrically if  $x_i$  is false. It is easy to check that this  $\mathbf{c}$  is a feasible network code.  $\square$

Figure 1: Reduction of 3-SAT to demand  $\tau = (T_1, \emptyset, T_3)$ .

**Corollary 3.3.** *For the general case with  $k > 2$  layers and demand  $\tau = (T_1, \emptyset, \dots, \emptyset, T_k)$ , we get NP-completeness with a slight modification of this proof, namely we add  $k - 3$  new  $sC_i$  and  $sd_j$  arcs.*

**Theorem 3.4.** *Given a network  $D$  and a demand  $\tau = (T_1, T_2)$  it is NP-hard to find a maximal cardinality subset  $T'_1$  of  $T_1$ , so that for  $\tau' = (T'_1, T_2)$  there exists a feasible network code.*

*Proof.* We prove the theorem by reducing the Vertex Cover problem. Let  $G = (W, E)$  denote an instance of this problem. For every vertex  $w \in W$  we add a receiver  $t_w \in T_1$  with an arc  $st_w$ , while for every edge  $uv \in E$  we add a receiver  $t_{uv} \in T_2$  with arcs  $t_u t_{uv}$  and  $t_v t_{uv}$ . For a given network code  $\mathbf{c}$ , a receiver node  $t_w$  can decode the first layer if and only if the height of the code on  $st_w$  is one. A receiver node  $t_{uv}$  can decode both layers, if on at least one entering arc the code has height two. Let  $T'_1 \subseteq T_1$  denote the set of nodes  $t_w, w \in W$  for which the arc  $st_w$  has height 2. It is easy to conclude that if the code is feasible for demand  $(T_1 \setminus T'_1, T_2)$  then  $T'_1$  is a vertex cover. Conversely, from a vertex cover  $T_0 \subseteq W$  we get a feasible network code for demand  $\tau_0 = (T_1 \setminus T_0, T_2)$  with height 2 on arcs incident to nodes in  $T_0$ , because for a fieldsize large enough ( $q \geq |W|$ ), all arcs with height two entering the same receiver can be chosen to be independent.  $\square$

As a minimal mixed (vertices and edges) cover of the edges may be supposed to contain only vertices, we also get the following.

**Corollary 3.5.** *Given a network  $D$ , a demand  $\tau = (T_1, T_2)$  and a number  $K$ , it is NP-complete to decide whether there exists a network code satisfying at least  $K$  requests.*

## 4 Limits on the Heights

**Definition 4.1.** A function  $f : A \rightarrow \{1, 2, \dots, k\}$  is called a **limit** on the arcs. An arc  $a$  with  $f(a) = i$  is an  **$i$ -valued** arc. A limit  $f$  is **realizable** for a proper demand  $\tau$ , if there is a feasible network code  $\mathbf{c}$  over some finite field, which has height  $H_{\mathbf{c}}(a) = f(a)$  for all  $a \in A$ .

### 4.1 Realizable limits for two layers

For two layers ( $k = 2$ ) the realizable limits can be characterized.

**Theorem 4.2.** *A limit  $f : A \rightarrow \{1, 2\}$  is realizable for a proper demand  $\tau = (T_1, T_2)$ , if and only if for all arcs  $uv \in A$ ,  $u \neq s$*

1. if  $f(uv) = 2$ , then  $\exists wu \in A : f(wu) = 2$ ,
2. if  $f(uv) = 1$ , then either  $\exists wu \in A : f(wu) = 1$ , or  $\lambda(s, u) \geq 2$ , and moreover
3. for any receiver  $t \in T_1$  with  $\lambda(s, t) = 1$ , there is a 1-valued arc entering  $t$ , and
4. for any  $t \in T_2$  there is a 2-valued arc entering  $t$ .

*Proof.* Let  $U \subseteq V \setminus \{s\}$  denote the set of special non-receiver nodes, where a node  $u$  is special, if all entering arcs are 2-valued, but it has a 1-valued outgoing arc (by Property 2, we know that  $\lambda(s, u) \geq 2$ ). The set of receiver nodes  $t \in T$  for which  $\lambda(s, t) = 1$  is denoted by  $T'_1$ . As  $\tau$  is proper, for each node in  $T'_2 = U \cup T \setminus T'_1$  there exist two arc disjoint paths from  $s$ . Using the basic network coding algorithm of Jaggi et al. [2] for receiver set  $T'_2$  one gets a network code  $\mathbf{c}$  feasible for the demand  $\tau_2 = (\emptyset, T'_2)$ . If the field size  $q$  is greater than  $|T'_2|$ , the code can be chosen to have height two on every arc, that is, the coefficient of  $\mathbf{e}_2$  is nonzero. In order to be feasible for the original demand  $\tau = (T'_1, T'_2)$ , we modify  $\mathbf{c}$  the following way: for every arc  $uv$  with  $f(uv) = 1$  set  $\mathbf{c}(uv) = (1, 0)$ .

We are left to prove that  $\mathbf{c}$  remains a network code with the linear combination property, and becomes feasible for demand  $\tau$ .

The span of the incoming vectors can only change at nodes which have only 1-valued incoming arcs, but in this case it has also only 1-valued outgoing arcs, so the network code has the linear combination property (note that in special nodes the span of the incoming vectors remains two-dimensional). Using Properties 3 and 4, the code becomes feasible for demand  $\tau$ .  $\square$

### 4.2 Limiting algorithm for two layers

There are proper demands for which no feasible network code exists.

**Definition 4.3.** For nodes  $u, v$  in a digraph  $D = (V, A)$ , a set  $X \subseteq V$  is an  $\bar{u}v$  **set** if  $v \in X$  but  $u \notin X$ . For a set  $X$  of nodes let  $\varrho(X)$  denote the number of entering arcs of  $X$ . A set  $X$  not containing  $s$ , and having  $\varrho(X) = i$  is called an  **$i$ -set**.

Note that by Menger's theorem,  $\lambda(s, v)$  equals the minimum of  $\varrho(X)$ , where  $X$  is an  $\bar{s}v$  set.

**Claim 4.4.** *Let  $v \in V \setminus \{s\}$ ,  $\lambda(s, v) = l$ , and  $X, Y$  two  $l$ -sets with  $v \in X \cap Y$ . Then  $X \cup Y$  is also an  $l$ -set.*

*Proof.* As  $\varrho(X \cup Y) + \varrho(X \cap Y) \leq \varrho(X) + \varrho(Y)$ , and  $\varrho(X \cup Y), \varrho(X \cap Y) \geq l$ , the claim follows.  $\square$

From the claim we get that for every vertex  $v \in V \setminus s$  there is a unique maximal  $\lambda(s, v)$ -set containing  $v$ .

Given a proper demand  $\tau = (T_1, T_2)$ , the following algorithm gives a realizable limit for  $\tau' = (T_1, T'_2)$  where  $T'_2$  is the unique maximal subset of  $T_2$ , such that a feasible network code for  $\tau'$  exists. As a byproduct, it also decides whether demand  $\tau$  can be satisfied or not. Having this limit, one can easily get a feasible network code for  $\tau'$  by the lines of the previous subsection. We remark that this code will also be feasible for  $\tau'' = (T_1 \cup (T_2 \setminus T'_2), T'_2)$ , in other words every receiver will get at least the base layer. We will also prove, that any fieldsize  $q > |T_1| + |T_2|$  will be enough for this network code.

Let  $\{Z_i\}$  be the maximal 1-sets which contain at least one node from  $T$ . Let  $I(Z_i)$  denote the set of arcs with head or tail in  $Z_i$ .

**Claim 4.5.** *The sets  $Z_i$  are pairwise disjoint and so are the sets  $I(Z_i)$ .*

*Proof.* If  $Z_i$  and  $Z_j$  are two different maximal sets of this property, Claim 4.4 implies that if they are not disjoint, or if there exists  $uv \in A$ ,  $u \in Z_i$ ,  $v \in Z_j$ , then  $Z_i \cup Z_j$  would be a bigger set with this property.  $\square$

Let  $Z$  denote the set of nodes not reachable from  $s$  in  $D' = (V, A \setminus \bigcup_i I(Z_i))$ . It is obvious that if every receiver in  $T$  can decode the first layer, then no receiver in  $Z$  can decode two layers. Let  $T'_2 = T_2 \setminus Z$ . For an arc  $uv \in A$ , let the limit  $f(uv)$  be the following. If  $uv \in I(Z)$ , then  $f(uv) = 1$ , otherwise let  $f(uv) = 2$ .

**Theorem 4.6.** *Limit  $f$  is realizable for  $\tau'' = (T_1 \cup (T_2 \setminus T'_2), T'_2)$ . In addition, any finite field of size  $q > |T_1| + |T_2|$  can be chosen for the network code.*

*Proof.* By the definition of  $Z$ , it is clear that Constraint 1 of Theorem 4.2 is fulfilled. Suppose that  $f(uv) = 1$  for an arc with  $u \neq s$  and there are no 1-valued arcs entering  $u$ . We need to prove that  $\lambda(s, u) \geq 2$ .

Suppose that this is not the case, thus there is an  $\bar{s}u$  set  $X \subset V$  with  $\varrho(X) = 1$ . Since  $uv \in I(Z)$  but none of the arcs entering  $u$  is in  $I(Z)$ , it follows that  $v \in Z$  and  $u \notin Z$ . Hence  $v \in Z_i$  for some  $i$ , but then  $X \cup Z_i$  would be a subset with in-degree one, contradicting the maximality of  $Z_i$ .

For the second statement, note that  $|U| \leq |T'_1|$ , because for a 1-valued arc  $uv$  if all arcs  $uv \in A$  have value two, then  $uv$  is the unique entering arc of a set  $Z_i$ , which contains a receiver  $t \in T$ . There can be at most  $|T'_1|$  such arcs, hence it is enough to have a field size greater than  $|T_1| + |T_2|$ .  $\square$

The proof is based on the estimation that  $|U| \leq |T'_1|$ . We leave the details for the full version.

We note that this algorithm has a more-or-less obvious implementation in time  $O(|A|)$  using BFS. We do not detail it here, because a more general distributed algorithm given in the next section will also do the job.

## 5 Three layers

Everything in this section is easily extendable for an arbitrary number of layers, but due to page limit, we detail the ideas only for three layers, and leave the generalization to the full version.

### 5.1 Realizable limits for three layers

For three layers we can give a sufficient condition for a limit to be realizable. We will consider **extensions** of a given limit  $f : A \rightarrow \{1, 2, 3\}$  to  $\hat{f} : (A \cup V) \rightarrow \{0, 1, 2, 3\}$ . We fix  $\hat{f}(s) = 3$ .

**Definition 5.1.** A path  $P$  with arcs  $a_1, a_2, \dots, a_r$  is called **monotone**, if  $f(a_1) \leq f(a_2) \leq \dots \leq f(a_r)$ . We define for such a monotone path  $\min(P) = f(a_1)$  and  $\max(P) = f(a_r)$ .

**Definition 5.2.** Let  $v \in V \setminus \{s\}$  be a node, and  $\hat{f}$  be an extended limit. An  **$i$ -fan** of  $v$  consists of  $i$  pairwise arc-disjoint non-trivial (i.e., containing at least one arc) monotone paths  $P_1, \dots, P_i$ , where for all  $j \leq i$  we have  $j \leq \min(P_j) \leq \max(P_j) \leq i$ , and  $P_j$  begins at a node  $v_j$  with  $\hat{f}(v_j) \geq \min(P_j)$ .

**Definition 5.3.** Given a limit  $f$ , an extension  $\hat{f}$  is called a **realizable extension**, if for every node  $v \in V \setminus \{s\}$  with  $i = \hat{f}(v) > 0$  there exists an  $i$ -fan of  $v$ ; and moreover, for every arc  $vw$ , either  $f(vw) \leq \hat{f}(v)$ , or there exists an incoming arc  $uv$  with  $f(uv) = f(vw)$ .

First we present a theorem about realizable limits. Next we show how one can algorithmically compute the maximal realizable extension of a limit. This algorithm also checks whether the given limit satisfies the conditions of the theorem.

Recall that a limit  $f$  is realizable for a proper demand  $\tau$ , if there is a feasible network code  $\mathbf{c}$  over some finite field, which has height  $H_{\mathbf{c}}(a) = f(a)$  for all  $a \in A$ .

In the classical algorithm of Jaggi et al. [2] they construct a feasible network code for a demand  $\tau = (\emptyset, \dots, \emptyset, T_k)$  by fixing  $k$  arc-disjoint paths to every receiver and constructing the network code on the arcs one by one, in the topological order of their tails, maintaining that the span of the last executed arcs on the fixed  $k$  paths remain the whole  $k$ -dimensional vector space. Their algorithm can be easily generalized for multi-layer demands.

**Theorem 5.4.** *Given a limit  $f : A \rightarrow \{1, 2, 3\}$  and a proper demand  $\tau = (T_1, T_2, T_3)$ , this limit is realizable, if it has a realizable extension  $\hat{f}$ , so that for every receiver node  $t \in T$ ,  $\hat{f}(t)$  is at least the request of  $t$ .*

*Proof.* If a receiver node can decode the first  $i$  layers then it can also send any linear combination of these layers. Let us call an arc  $uv$  **free** if  $f(uv) \leq \hat{f}(u)$ . Note that every starting arc of a path in a fan is free. Moreover, if  $v$  has an  $i$ -fan then it also has an  $i$ -fan with exactly one free arc on each path, because a path  $P_j$  of a fan can be replaced by the subpath  $P'_j$  from the last free arc on  $P_j$ . Let us fix such a fan for every node with nonzero extended limit. We modify the algorithm of Jaggi et al. [2] the following way: on free arcs we define the network code in increasing order of the limits on the arcs. Since the paths in a fan satisfy that  $\min(P_j) \geq j$ , we can define the network code so that for every fan  $\dim\langle \mathbf{a}_1, \dots, \mathbf{a}_j \rangle = j$  for all  $1 \leq j \leq i$ , where  $\mathbf{a}_j$  is the first arc on path  $P_j$ . On non-free arcs we define the network code in a topological order of their tails. When constructing the network code on a non-free arc  $uv, u \neq s$ , we maintain that for every  $i$ -fan which contains  $uv$ , the span of the last executed arcs on the  $i$  paths remain the first  $i$  layers. We use the following lemma to prove that this is possible.

**Lemma 5.5.** [2] *Let  $n \leq q$ . Consider pairs  $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$  with  $\mathbf{x}_i \cdot \mathbf{y}_i \neq 0$  for  $1 \leq i \leq n$ . There exists a linear combination  $\mathbf{u}$  of vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  such that  $\mathbf{u} \cdot \mathbf{y}_i \neq 0$  for  $1 \leq i \leq n$ .*

If vectors  $\mathbf{v}_1, \dots, \mathbf{v}_l$  span the subspace of the first  $l$  layers, for every  $\mathbf{v}_i, 1 \leq i \leq l$  there is a vector  $\mathbf{y}_i$  in this subspace with  $\mathbf{v}_j \cdot \mathbf{y}_i = 0, i \neq j$  and  $\mathbf{v}_j \cdot \mathbf{y}_j \neq 0$ . We call  $\mathbf{y}_j$  a **control vector** of  $\mathbf{v}_j$ . Let  $P_1, \dots, P_l$  denote paths which contain  $uv$  with preceding arcs  $w_1u, \dots, w_lu$  in fans  $F_1, \dots, F_l$ , respectively, and let  $\mathbf{y}_i, 1 \leq i \leq l$  be the control vector of  $\mathbf{c}(w_iu)$  for the set of last coded vectors on the other paths of  $F_i$ . Applying Lemma 5.5 with  $\mathbf{x}_i = \mathbf{c}(w_iu)$  and  $\mathbf{y}_1, \dots, \mathbf{y}_l$  we get that there is a good linear combination  $\sum_{i=1}^l \lambda_i \mathbf{c}(w_iu)$ . Let  $\mathbf{c}(uv)$  be this sum. The height of  $\mathbf{c}(uv)$  will be at most the height of arcs  $w_iu$ , hence it remains under the given limit, because all  $P_j$ 's are monotone. Let  $wu$  be an entering arc with  $f(wu) = f(uv)$ . Since  $uv$  is a non free arc, such an arc exists. By adding  $\mathbf{x}_{l+1} = \mathbf{c}(wu), \mathbf{y}_{l+1} = e_{f(uv)}$  we can achieve that the height of  $uv$  is  $f(uv)$ .  $\square$

**Claim 5.6.** *If a limit has a realizable extension, then it has a unique maximal realizable extension.*

*Proof.* Let  $f^+$  be a realizable extension for which  $\sum_{v \in V} f^+(v)$  is maximal and assume indirectly that there exists another realizable extension  $f'$  and a node  $v$  for which  $f'(v) > f^+(v)$ . We can assume that  $v$  is the first such node in a topological order. Clearly, increasing  $f^+$  on  $v$  to  $f'(v)$  would also give a realizable extension, because the  $i$ -fan of  $v$  is also an  $i$ -fan for extended limit  $f^+$ .  $\square$

**Theorem 5.7.** *The maximal realizable extension of a limit can be determined algorithmically.*

*Proof.* We calculate the maximal realizable extension in a topological order of the nodes. For a node  $v \in V \setminus s$  and a given limit value  $i$ , let  $D_{v,i} = (V', A')$  denote the following auxiliary graph of  $D$ : We delete all arcs with limit greater than  $i$ . We add  $i$  extra nodes to the digraph:  $t_1, \dots, t_i$  with  $2i - 1$  extra arcs:  $st_j, 1 \leq j \leq i$  and

$t_j t_{j+1}$ ,  $1 \leq j \leq i - 1$ . For every node  $u$  before  $v$  in the topological order we change the tail of every outgoing arc  $uw$  from  $u$  to  $t_{f(uw)}$  if  $\hat{f}(u) \geq f(uw)$ .

**Lemma 5.8.** *There exists an  $i$ -fan to  $v \in V$  if and only if  $\lambda_{D_{v,i}}(s, v) = i$ .*

*Proof.* Note that a monotone path  $P$  to  $v$  in  $D$ , with exactly one free arc, corresponds to a path in  $D_{v,i}$  starting from  $t_{\min(P)}$  and vice versa. Hence an  $i$ -fan corresponds to  $i$  paths in  $D_{v,i}$ , each starting from a node  $t_j$  for some  $j$ . Suppose indirectly that  $\lambda_{D_{v,i}}(s, v) < i$ , that is, there exists an  $\bar{sv}$  set  $X \subseteq V'$  with  $\varrho(X) < i$ . Since  $\lambda_{D_{v,i}}(s, t_i) = i$ ,  $t_i \notin X$ . Let  $j$  denote the greatest integer for which  $t_j \in X$ . Since for an  $i$ -fan at least  $i - j$  paths in the fan have value at least  $j + 1$ , paths in  $D_{v,i}$  corresponding to paths of the fan enter  $X$  on at least  $i - j$  arcs. Also, there are  $j$  paths to  $t_j$  in  $D_{v,i}$  using arcs between  $s$  and  $t_1, \dots, t_j$  only, which are disjoint from the arcs of the fan. Hence there are at least  $i$  arcs entering  $X$ , contradicting the assumption.

To prove the other direction let  $P_1, P_2, \dots, P_i$  be  $i$  arc-disjoint  $sv$  paths in  $D_{v,i}$ . Note that  $\{t_1, \dots, t_i\}$  is a cut set in  $D_{v,i}$  hence a path  $P_j$  must go through at least one of them. Since  $\varrho(\{t_1, \dots, t_j\}) = j$ , at least  $i - j + 1$  paths go through the set  $\{t_{j+1}, \dots, t_i\}$ , which correspond to paths in  $D$  with value at least  $j + 1$ .  $\square$

The maximal possible value of  $\hat{f}(v)$  is the maximal  $i$  for which there exist an  $i$ -fan of  $v$ .  $\square$

Lemma 5.8 shows that the existence of a fan is equivalent with a connectivity requirement in an auxiliary graph. With a slight modification of the following algorithm one can get a distributed algorithm for determining the maximal realizable extension of a fan.

## 5.2 A distributed connectivity algorithm

Goals: we are going to give a distributed, linear time algorithm for the following problems:

- Determine  $\lambda(s, v)$  for all  $v$ , but if it is  $\geq 3$  then only this fact should be detected.
- For each  $v$  with  $\lambda(s, v) = 1$  determine the incoming arc of the unique maximal 1-set containing  $v$ .
- For each  $v$  with  $\lambda(s, v) = 2$  determine the incoming arcs of the unique maximal 2-set containing  $v$ .

We assume that  $*$  is a special symbol which differs from all arcs.

During the algorithm each node  $v$  (except  $s$ ) waits until it hears messages along all incoming arcs, then it calculates  $\lambda(s, v)$ , and the 3 messages  $m_1(v), m_2(v), m_3(v)$  it will send along all outgoing arcs.

The algorithm starts with  $s$  sending  $m_1(s) := m_2(s) := m_3(s) := *$  along all outgoing arcs.

We need to describe the algorithm for an arbitrary node  $v \in V \setminus \{s\}$ . First  $v$  waits until hearing the messages on the set of incoming arcs denoted by  $IN(v) =$

$\{a_1, \dots, a_r\}$ . When on an arc  $a_i$  it hears a  $*$ , it replaces it by  $a_i$ . Let the messages arrived (after these replacements) on arc  $a_i$  be  $m_1^i, m_2^i, m_3^i$ . Then  $v$  examines the set  $M_1(v) = \{m_1^i\}_{i=1}^r$ . If  $|M_1(v)| = 1$  then  $v$  sets  $\lambda(s, v) := 1$  and  $m_1(v) := m_1^1$ , otherwise it sets  $m_1(v) := *$ .

Next  $v$  examines the set  $M_2(v) = \bigcup_{i=1}^r \{m_2^i, m_3^i\}$ . If  $|M_2(v)| = 2$  then it sets  $\{m_2(v), m_3(v)\} = M_2(v)$ , and if  $\lambda(s, v)$  was not set to 1 before, it sets it to 2.

Let us call an entering arc  $a_j$  **important** for  $v$ , if  $m_1^j \notin \bigcup_{1 \leq i \leq r, i \neq j} \{m_2^i, m_3^i\}$ , and let  $I_v$  denote the set of important arcs for  $v$ . If  $|M_2| > 2$ , then  $v$  next examines the set  $(v) = \bigcup_{i \in I_v} \{m_2^i, m_3^i\}$ , and if  $|(v)| = 2$ , then it makes the same steps with  $(v)$  as described before with  $M_2(v)$ .

Finally, if both  $|(v)|$  and  $|M_2(v)|$  are greater than 2 and  $\lambda(s, v)$  was not set to 1,  $v$  examines  $M_1(v)$  again, and if  $|M_1(v)| \leq 2$ , then it sets  $\{m_2(v), m_3(v)\} = M_1(v)$ , and it sets it to 2.

If they were not set before, let  $m_2(v) := m_3(v) := *$  and  $\lambda(s, v) = 3$ .

An  **$sv$ -cut** is a set of arcs for which intersects every  $sv$  path.

**Claim 5.9.** *Let  $v \in V \setminus \{s\}$ . Each of the sets  $M_1(v)$ ,  $(v)$  and  $M_2(v)$  contains an  $sv$ -cut.*

*Proof.* For an arc  $a$ , let us call an arc set an  $a$ -arc-cut if it intersects every directed path from  $s$  ending with  $a$ . Note that the arc  $a$  itself is an  $a$ -arc-cut and the union of arc-cuts for all the entering arcs of a node  $v$  form an  $sv$ -cut. Also, for an arc  $uv$  an  $su$ -cut forms a  $uv$ -arc-cut, too. To prove the claim, inductively we can assume that on an arc  $uv$  both  $m_1(u)$  and  $\{m_2(u), m_3(u)\}$  are either  $uv$ , or an  $sv$ -cut. In both cases they form a  $uv$ -arc cut, proving the claim.  $\square$

**Theorem 5.10.** *For every node  $v \in V \setminus \{s\}$ , the algorithm correctly calculates  $\lambda(s, v)$ . If  $\lambda(s, v) = 1$  then  $m_1(v)$  is the incoming arc of the unique maximal  $\bar{sv}$  set with  $\varrho(X) = 1$ . If for the arc  $uw$  entering this set  $X$  we have  $\lambda(s, u) = 2$ , then  $\{m_2(v), m_3(v)\} = \{m_2(u), m_3(u)\}$ . If  $\lambda(s, v) = 2$  then  $m_2(v), m_3(v)$  is the pair of incoming arcs of the unique maximal  $\bar{sv}$  set with  $\varrho(X) = 2$ .*

*Proof.* First suppose that  $\lambda(s, v) \geq 3$ . By Claim 5.9,  $|M_1(v)| \geq 3$ ,  $|(v)| \geq 3$ . and  $|M_2(v)| \geq 3$ . Consequently in this case node  $v$  correctly concludes  $\lambda(s, v) \geq 3$  and it will send  $*$ s as messages.

Now suppose  $\lambda(s, v) = 1$ , and let  $X$  denote the unique maximal set with  $s \notin X$ ,  $v \in X$ ,  $\varrho(X) = 1$ , and let  $uw$  be the unique arc entering  $X$ . In this case clearly  $m_1(w) = uw$  (otherwise  $m_1(w)$  would be an arc  $e$  entering another set  $Y$  with  $u \in Y$  and  $\varrho(Y) = 1$ , but then  $X \cup Y$  would be a bigger set with one incoming arc). It is easy to see that now along every arc inside  $X$  the first message is also  $uw$ , so only this message arrives at  $v$  as first message and then  $v$  correctly sets  $\lambda(s, v) = 1$ . Also, if  $\lambda(s, u) = 2$ , then inductively we may assume that  $|\{m_2(u), m_3(u)\}| = 2$  hence  $M_2(z)$  remains this set for every node only reachable from  $u$ , including  $v$ .

Finally suppose  $\lambda(s, v) = 2$ , and let  $X$  denote the unique maximal set with  $s \notin X$ ,  $v \in X$ ,  $\varrho(X) = 2$ , and let  $uw$  and  $u'w'$  be the two arcs entering  $X$ . Note that  $\lambda(s, u), \lambda(s, u') > 1$ , otherwise  $X$  would not be maximal. That is,  $m_1(u) = m_1(u') = *$ . By Claim 5.9,  $|M_1(v)| \geq 2$ , so  $v$  does not set  $\lambda(s, v)$  to one.

As  $D$  is acyclic with a unique source  $s$ , the subgraph of  $D$  spanned by  $X$  either contains one source, say  $w$ , or two sources:  $w$  and  $w'$ .

*Case I*  $w = w'$ . As  $w$  is the source of  $G[X]$ , we have  $\varrho(w) = 2$ , so  $|M_1(w)| = 2$  and  $m_2(w) \neq *$  and  $m_3(w) \neq *$ . Therefore every node  $x$  inside  $X$  has  $M_2(x) = \{m_2(w), m_3(w)\}$ . As  $|M_2(v)| = 2$ ,  $v$  sets  $\lambda(s, v) = 2$ .

*Case II*  $w \neq w'$ . Let  $X_1$  denote the set of vertices  $x \in X$  only reachable from one of  $w$  and  $w'$ . It follows that  $\lambda(s, x) = 1$  for all  $x \in X_1$  hence every  $x \in X_1$  has  $M_1(x) = \{uw\}$  or  $\{u'w'\}$ . For a node  $v \notin X_1$  with all parents in  $X_1$   $M_1(v) = \{uw, u'w'\}$  and inductively for every other node  $v$   $(v) = \{uw, u'w'\} \subseteq M_2(v)$ , since all entering arcs not coming from  $X_1$  are important for  $v$ .  $\square$

### 5.3 Heuristic for 3 or more layers

In this subsection we give an algorithm for three layers. We prove that the algorithm sends the first layer to every receiver and within this constraint the unique maximal set of receivers get at least two layers, while some receivers may get three layers.

*Step 1* Let  $W_1$  denote the union of maximal 1-sets which contain at least one node from  $T$ . In Section 4 it was proved that if all receivers get the first layer, a receiver  $v$  cannot get more than one layer if and only if it is cut by  $W_1$  from  $s$ , that is, if there is no directed path from  $s$  to  $v$  in  $V \setminus W_1$ . Let  $\overline{W}_1 \supseteq W_1$  denote the set of nodes cut from  $s$  by  $W_1$ . We set  $T_1 = T \cap \overline{W}_1$ . We define a set of **pseudo receivers**  $U$  which contains nodes not in  $\overline{W}_1$  but having an outgoing arc entering  $\overline{W}_1$ .

*Step 2* Similarly to the first case, let  $W_2$  denote the maximal 2-sets which contain a receiver or a pseudo receiver. Let  $\overline{W}_2 \subseteq V \setminus \overline{W}_1$  denote the set of nodes only reachable from  $s$  through  $\overline{W}_1 \cup W_2$ . We set  $T_2 = (T \cup U) \cap \overline{W}_2$ .

Note that for determining sets  $\overline{W}_1$  and  $\overline{W}_2$  we can use the distributed algorithm presented in Subsection 5.2.

*Step 3* We define a limit  $f$  on  $D$  which is 1 on  $I(\overline{W}_1)$ , 2 on  $I(\overline{W}_2) \setminus I(\overline{W}_1)$  and 3 otherwise. We proceed on the nodes of  $T^* = (U \cup T) \setminus T_1$  in a fixed a topological order and decrease  $f$  on some arcs from 3 to 2. Let  $v$  denote the next node to be processed. We take a cost function  $c : A \rightarrow \{0, 1\}$  which is 1 on 3-valued arcs and 0 everywhere else. Since  $v \notin \overline{W}_1$ , there are two arc-disjoint paths  $P_1$  and  $P_2$  from  $T \cup U \setminus \{s\}$  to  $v$  so that  $P_2$  does not start in  $T_1$ . Moreover, it can be assumed that the inner nodes of these paths do not intersect  $T^*$ .

*Case I*  $P_1$  can also be chosen not to start in  $T_1$ . Let us take a minimum cost pair of paths  $P_1 \cup P_2$  described above according to the cost function  $c$ . Then we decrease  $f$  on the 3-valued arcs of  $P_1$  and  $P_2$ .

*Case II*  $P_1$  starts in  $T_1$ . Again, we take a minimum cost  $P_1 \cup P_2$  according to the cost function  $c$ . Then we decrease  $f$  on the 3-valued arcs of  $P_1$  and  $P_2$ .

*Step 4* Finally, we check in the topological order of the nodes if every 3-valued outgoing arc has a 3-valued predecessor, and if not, we decrease its value to 2.

**Theorem 5.11.** *The limit  $f$  achieved has a realizable extension for demand  $\tau = (T_1, T_2', T_3')$  for which  $T_2 \subseteq T_2'$  and  $(T_2' \cup T_3') \supseteq T \setminus T_1$ .*

## 6 Conclusion

In this paper we investigated the multi-layered multicasting problem proposed by Kim et al. [3]. We proved NP-hardness for some very special cases of the problem, including demand  $\tau = (T_1, T_2)$ , if we want to maximize the number of satisfied receivers. For two layers we gave a network coding algorithm which is optimal if the task is to send at least one layer to every receiver and two layers to as many receivers as possible. For three layers we gave a sufficient condition for a limit to be realizable and showed that this condition can be checked algorithmically. Also, we presented a heuristic for three layers, which not only ensures that all terminals receive the base layers, but also carries the second layer to the maximum number of receivers.

## References

- [1] Michelle Effros, *Universal multiresolution source codes*, IEEE Transactions on Information Theory **47** (2001), no. 6, 2113–2129.
- [2] Sidharth Jaggi, Peter Sanders, Philip A. Chou, Michelle Effros, Sebastian Egner, Kamal Jain, and Ludo M. G. M. Tolhuizen, *Polynomial time algorithms for multicast network code construction.*, IEEE Transactions on Information Theory (2005), 1973–1982.
- [3] MinJi Kim, Daniel Enrique Lucani, Xiaomeng Shi, Fang Zhao, and Muriel Médard, *Network coding for multi-resolution multicast*, INFOCOM, 2010, pp. 1810–1818.
- [4] Nachum Shacham, *Multipoint communication by hierarchically encoded data*, INFOCOM, 1992, pp. 2107–2114.
- [5] Raymond W. Yeung, Shuo-Yen Robert Li, Ning Cai, and Zhen Zhang, *Network coding theory - part i: Single source*, vol. 2, Foundations and Trends in Communications and Information Theory, no. 4, now Publishers Inc., 2005.