Maximum Network Flow of Multiple Description Codes

Bin Ma[†] Xiaolin Wu[‡]

Abstract— **Multiple description coding (MDC) is a powerful technique for robust real-time communications over packet switched lossy networks, such as the Internet, peer-to-peer, ad hoc, wireless, and sensor networks. This paper is concerned with optimal network flows of MDC packets from multiple sources to multiple receivers to maximize a weighted sum of fidelities achieved by all receivers. For general network topology and general fidelity function this problem is proven to be Max-SNP-hard, i.e., there is no polynomial-time algorithm to even approximate the optimal solution. However, the problem is tractable and algorithms are proposed for some practically important cases, such as when the MDC has equally important packets of a fixed size (e.g., Reed-Solomon coded packets), or/and if the network topology is a tree. A linear integer programming algorithm for arbitrary network topology is also proposed, which is quite practical for modestly sized networks although having exponential worst-case complexity.**

I. INTRODUCTION

Packet switched networks, such as the Internet, peer-to-peer, ad hoc, and diversity wireless networks, inevitably experience packet losses and delays in practice. Packet retransmission is undesirable either due to latency constraints in real-time applications or due to bandwidth economy or both. In contrast, transmission policies on a best-effort basis offer simpler, faster, and less expensive solutions, in which no acknowledge from the receiver is needed, nor is there guarantee that the data packets will arrive in order, or at all. This simple send-only machinery shifts the burden of reliability from the network protocols toward the communication codes. The need for more sophisticated codes to compensate for lesser network provisions has led to a proliferation of research literature on designing distributed source codes and joint source-channel codes for packet-switched networks and erasure channels (Ref. lists of [9] and this paper). Another elegant paradigm arising to the same challenge is network coding, in which the nodes of the network can perform coding operations [1], [5].

This paper deals with the rate-distortion optimized interaction between networking and coding. We are interested in how to optimize network flows for distributed source codes or joint source-channel codes, given the topology and edge capacities of a network, and under a linearly weighted fidelity criterion over the entire network. This work is related to network coding, but its perspective and approach are different from those of network coding as commonly known in the recent literature. We respect the physical limitations of the existing network infrastructure, and assumes that the nodes of the network can only forward and duplicate data received, just as the network routers do in reality. But even under this constraint, the optimal network flow of data packets in rate-distortion sense behaves very differently from maximum flow of commodities in network problems treated in classic operation research literature. Computationally, our problems are far more complex because the reconstruction fidelity achieved by a decoder is not additive of the fidelities offered by the individual data packets received.

We consider the network flows of general multiple description codes (MDC). MDC is a powerful technique for robust real-time networked communications, and in particular for enhanced QoS of networked multimedia streaming, including all forms of digital media: image [17], [13], video [11], audio [3], and graphics [2]. Research on MDC started in late seventies both as an engineering problem [12] and a theoretical investigation (the problem was first posed at the 1979 IEEE Information Theory Workshop by Wyner). The existing MDC techniques fall into two categories: distributed source coding and distributed joint source-channel coding. Multiple description quantization, in both scalar and vector variants [6], [8], [14], [18], [19], and correlating transforms [9] are in the first category. Also, multiresolution (or layered) source codes can be considered as a special type of MDC, and they fall into the first category since different layers can be transmitted via different paths from the source to destination. Uneven error protected (UEP) packetization of scalable source code streams by packet erasure codes, an intensively-studied MDC technique in recent years [4], [7], [13], [16], [17], belongs to the joint source-channel coding category.

For our general problem of optimizing the network performance in delivering MDC-coded contents, we do not distinguish between MDCs generated by distributed joint source-channel coding and by distributed source coding. It suffices to consider their commonality of coding a signal into a set of N co-descriptors (side descriptions) $X = \{x_1, x_2, \cdots, x_N\}$. However, the reader is reminded of an important difference between the UEP-based MDC and the distributed source coding. The former creates co-descriptors of equal importance, whereas the latter, such as multiple description quantization and correlating transforms, does not in general. This difference will affect the definition of objective function in the underlying optimization problem, although not the problem formulation.

One way of maintaining the quality of service, at times of network congestions and server breakdowns, is to afford a client multiple accesses to a given content, by distributing co-descriptors of an MDC-coded signal to different servers in the network. In this paper we consider a network of a fixed topology, modelled by a graph $G = \langle V, E\rangle$. The nodes in the set V correspond to servers, clients, and possibly relays on the network. The edges in the set E represent direct links between these nodes. Each edge has a capacity, reflecting the capacity of the underlying channel. Let $\mathcal{X}_k = \{x_{k,1}, x_{k,2}, \cdots, x_{k,n_k}\} \subset X$ be the subset of MDC co-descriptors at server k. Subsets $\mathcal{X}_k$ and $\mathcal{X}_j$ resided at servers k and j are in general not disjoint, $\mathcal{X}_k \cap \mathcal{X}_j \neq \emptyset$, for a desired degree of redundancy to guard

---

[†] Department of Computer Science, University of Western Ontario, London, Ontario, Canada N6A 5B7, bma@csd.uwo.ca.

[‡] Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4K1, xwu@mail.ece.mcmaster.ca.

against hostile network conditions. Any client in the network can decode the MDC-coded signal upon receiving a subset $\mathcal{X} \subset X$ of co-descriptors with fidelity $F(\mathcal{X})$. We are interested in maximizing the reconstruction quality of the signal given the network graph $G = \langle V, E \rangle$ and edge capacities, and given the subsets of MDC co-descriptors possessed by the servers. This optimization problem is called *maximum MDC network flow*. There are two versions of the maximum MDC network flow problem depending on whether the maximization is with respect to a single sink or to multiple sinks.

For the popular technique of UEP packetization of scalable source code stream using RS code, all MDC co-descriptors (packets) have the same size and same importance. Consequently, the fidelity function $F(\mathcal{X})$ only depends on the cardinality of $\mathcal{X}$ not the particular composition of $\mathcal{X}$. Maximizing reconstruction quality becomes equivalent to maximizing the number of received distinct packets given the network edge capacities. In a sister paper [20] we studied this problem, called it *rainbow network flow*. We showed that rainbow network flow problem is NP-hard if the goal is to maximize the total number of distinct co-descriptors received by multiple sinks from multiple sources. However, the single-sink version of the problem can be reduced to one of classic maximum network flow and hence is polynomially solvable.

In this paper we investigate the general maximum MDC network flow problem for arbitrary fidelity function $F(\mathcal{X})$, in which $F(\mathcal{X})$ accounts for not only the correlations between co-descriptors such as those of multiple description quantizers, but also the sequential decoding dependency between co-descriptors such as those in layered multi-resolution source coding. We will show that such a generalization makes the problem much harder.

The rest of the paper proceeds as follows. Sec. II formulates the general MDC network flow problem, followed by the proof of the hardness of the problem in Sec. III. After showing the MDC network flow problem to be MAX-SNP-hard, we turn our attention in Sec. IV to some cases for which a polynomial-time algorithm exists. The algorithmic study is then extended to so-called duplicable MDC network flow problem in Sec. V. Sec. VI offers an integer programming approach to solving the MDC network flow problem in general setting exactly, and Sec. VII concludes.

II. PROBLEM FORMULATION

The input of the *Maximum MDC Network Flow Problem* include a fixed network topology represented by directed graph $G = \langle V, E \rangle$, in which each edge $e \in E$ has a capacity $C(e) \geq 0$; an MDC code consisting of a set of N co-descriptors $X = \{x_1, x_2, \cdots, x_N\}$ of lengths $l_1, l_2, \cdots, l_N$; a set of source nodes $S_n \subset V$ that possess descriptor x_n, $1 \leq n \leq N$; and a receiver (sink) set $T \subset V$.

An MDC network flow consists of N sets (some may be empty) of directed paths, $P_1, \ldots, P_N$. A path p in a non-empty P_n, $1 \leq n \leq N$, carries a distinct descriptor x_n from a source node $s \in S_n$ to a receiver node $t \in T$. Let $\mathcal{X}_t = \{x_{t,1}, x_{t,2}, \cdots, x_{t,n_t}\} \subset X$ be the subset of co-descriptors carried by the MDC network flow to receiver $t \in T$, and $F(\mathcal{X})$ be the reconstruction fidelity achieved by receiving $\mathcal{X} \subset X$. The maximum MDC network flow problem is to

$$\max_{P_1,\ldots,P_N} \sum_{t \in T} w(t) F(\mathcal{X}_t) \tag{1}$$

subject to

$$\sum_{n=1}^{N} \sum_{\rho \in P_n, e \bowtie \rho} l_n \leq C(e), \quad \forall e \in E \tag{2}$$

where $e \bowtie \rho$ in the summation subscript means that edge e is on the path ρ, and $w(t)$ is a weighting function to prioritize different receivers in optimizing MDC network flows (by user fees, urgency, and etc.) In other words, we want to maximize the weighted sum of reconstruction fidelities over all receivers while making all paths collectively satisfy the capacity constraints.

Following [20], we say that different descriptors have different *colors*, and a path p that carries co-descriptor x_n to have color n.

Also, we define an *Undirected MDC Network Flow Problem* the same as MDC Network Flow Problem except that the edges and paths in the definition are undirected.

III. HARDNESS RESULTS OF MDC NETWORK FLOW PROBLEM

In [20], we proved that the rainbow network flow problem, a special case of the MDC network flow problem when the reconstruction fidelity $F(\mathcal{X}) = |\mathcal{X}|$, is NP hard. Thus, the MDC Network Flow problem is also NP-hard.

Moreover, we find that the MDC network flow problem is even "harder" than the rainbow network flow problem. Specifically, when there is only one sink node, the latter problem permits a polynomial-time solution as shown in [20], whereas the former problem remains NP-hard as we will prove in the rest of the section. In fact, we will prove a stronger complexity result that even the single-sink MDC network flow problem is Max-SNP-hard. Namely, there is a constant $\epsilon > 0$, such that no polynomial-time algorithm can approximate MDC network flow problem with ratio better than $1 + \epsilon$.

Our proof is based on the fact that the Max-2-SAT problem is Max-SNP-hard [15]. The definition of the Max-2-SAT problem is the following.

Max-2-SAT Let $x_1, x_2, \ldots, x_n$ be n boolean variables. Let y_i be 2-CNF (conjunctive normal form) of the boolean variables. Find a truth assignment of the variables that satisfies the maximum possible number of the clauses.

Theorem 1: MDC network flow problem is Max-SNP-hard even if there is only one sink node, all the lengths of descriptions are equal to one, and the network topology is a tree.

Proof: We construct an L-reduction [15] from Max-2-SAT to MDC network flow problem. Let $\mathcal{I}$ be an instance of the Max-2-SAT problem with n variables, $x_1, x_2, \ldots, x_n$ and m clauses y_i, $i = 1, 2, \ldots, m$.

Our constructed instance of MDC network flow problem has $2n$ descriptors, each corresponds to either x_i or $\bar{x}_i$. It also has $2n$ source nodes. The source nodes are grouped into m pairs, s_i and t_i, $i = 1, 2, \ldots, n$. Each s_i has the descriptor corresponding to x_i, and each t_i has the descriptor corresponding to $\bar{x}_i$.

Then all the nodes are connected as in Figure 1. There is only one sink node t. Each descriptor has length one. And the capacity of each edge is also one. This graph ensures that only one of the two descriptors x_i and $\bar{x}_i$ can reach the sink t.

Therefore, if there are exactly n descriptors arriving t, we can accordingly get an assignment to the variables x_i ($i = 1, 2, \ldots, m$) as follows

$$x_i = \text{true} \quad \Leftrightarrow \quad \text{descriptor } x_i \text{ arrives } t.$$

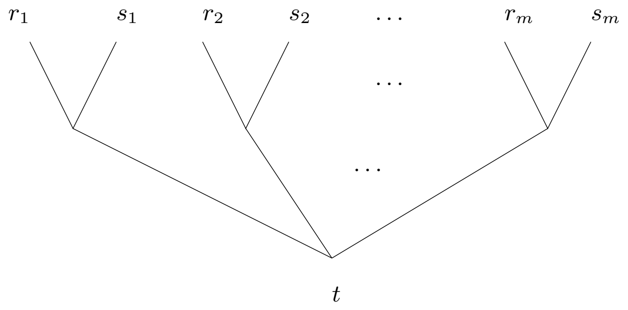


Fig. 1. The topology of the constructed MDC network flow in the proof of Theorem 1.

The reconstruction fidelity achieved by the n descriptors is then defined by the number of clauses satisfied by this assignment.

If there are less than n descriptors arriving t, we define the reconstruction fidelity to be 0.

Thus, we constructed an instance, $\mathcal{J}$, of MDC network flow problem. As discussed above, a solution of the constructed $\mathcal{J}$ with reconstruction fidelity M can be used to construct a solution of $\mathcal{I}$ with M satisfied clauses; and vice versa. Therefore, the reduction we have shown is an L-reduction.

Because Max-2-SAT is Max-SNP-hard, MDC network flow problem is also Max-SNP-hard. ■

It is easy to see that the reduction used in the proof of Theorem 1 also works when the graph is undirected. Therefore, we have

Theorem 2: Undirected MDC network flow problem is Max-SNP-hard even if there is only one sink node, all the lengths of descriptions are equal to one, and the network topology is a tree.

IV. POLYNOMIAL-TIME ALGORITHM FOR MDC NETWORK FLOW ON TREE TOPOLOGY

Although MDC network flow problem is NP-hard in general, there are cases when polynomial time algorithms exist. If the network topology is a tree and there is a constant number of MDC co-descriptors that is independent of the network size (typically true in the practice of MDC coding), then the RNF problem can be solved in polynomial time. The tree topology is common in local area networks (LAN) and in sensor networks [10].

We develop an MDC network flow algorithm first for binary trees, and then extend it to arbitrary trees.

First clarify what we mean by a tree topology when the graph is directed. Let $G = \langle V, E\rangle$ be a directed graph. We can construct an undirected graph $G' = \langle V, E'\rangle$ as follows: $(u, v) \in E'$ if and only if $(u, v) \in E$ or $(v, u) \in E$. If G' is a tree, we say that G has a tree topology. Note that our definition of tree topology is undirectional, allowing two-way communications on any network edge. This expands the applicability of our network model. Fig. 2(a) shows an example of a directed graph of a tree topology.

Arbitrarily select a leaf v from G'. Add a dummy vertex v^* and a dummy edge (v, v^*) into G. Then G becomes a rooted binary tree with v being the root. Fig. 2 illustrates the corresponding effects on the directed graph G. Consequently, each vertex $u \in V$ is either an internal node having two child nodes, or a leaf of no descendants. The subtree rooted at a node $u \in V$ can also be naturally defined.

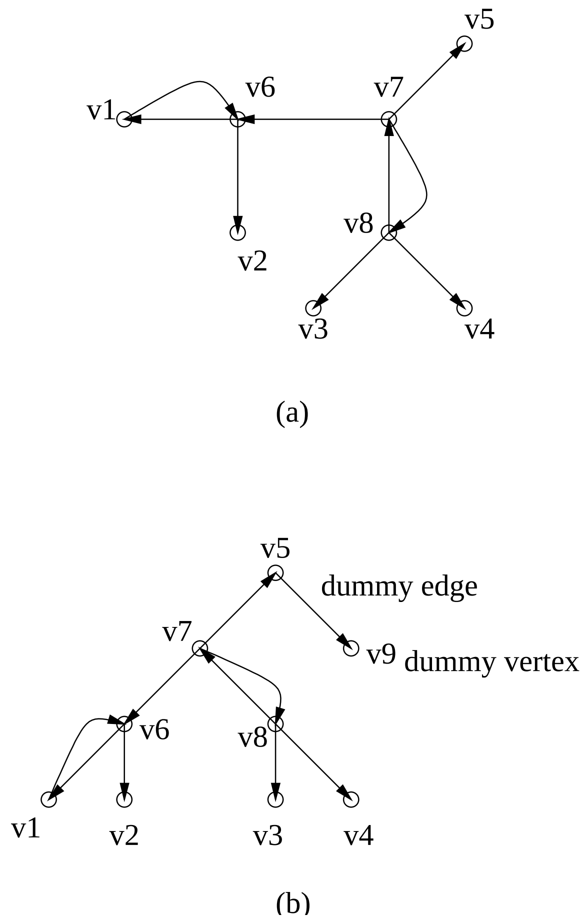


Fig. 2. (a) An example of a flow network that has a tree topology. (b) By adding a dummy vertex and a dummy edge, the tree topology can be regarded as rooted.

For each internal node u, let u' be the parent node of u. Suppose there are x_j paths of color j that pass both u and u' in the optimal flow. Obviously, it is useless to have paths of the same color to pass both u and u' from different directions. In that case, we can easily modify the optimal solution so that all the paths that pass both u and u' will have the same direction between u and u'. Therefore, without loss of generality, we assume that all the x_j paths pass both u and u' in the same direction. If they go from u' to u, we let $f_j = x_j \geq 0$; Otherwise, we let $f_j = -x_j \leq 0$. Then we can unambiguously say that there are f_j paths flowing into u from its parent.

Our algorithm uses the Dynamic Programming technique. Let $DP(u, f_1, f_2, \ldots, f_k)$ be the maximum total fidelity of the sinks in the subtree rooted at u, with the condition that there are f_j color-j paths flowing into u from the parent. Let v, w be the two children of u. Dynamic Programming relies on a recurrence relation to compute $DP(u, f_1, f_2, \ldots, f_k)$. This is accomplished by considering the following four different cases of u. Let u' be the parent of u.

Case 1. u is a leaf node but not a sink.

In this case, we ban any co-descriptor to flow into u. That is, $f_i \leq 0$. For every co-descriptor x_i for which u is a source, we allow $f_i < 0$; otherwise, f_i must be 0. At the same time, $f_i < 0$ must satisfy the capacity constraint at the directed edge (u, u'). To summarize, $f_1, f_2, \ldots, f_k$ must satisfy the following conditions

1) $f_i \le 0$ for $u \in S_i$,
2) $f_i = 0$ for $u \notin S_i$,
3) $\sum_{j:f_j<0} -f_j l_j \le C(u,u')$,

Let $\mathcal{F}_1$ be the set of vectors $(f_1, f_2, \ldots, f_k)$ that satisfy the above mentioned conditions. Any feasible $(f_1, f_2, \ldots, f_k)$ must be in $\mathcal{F}_1$. Because u is not a sink, the contribution of u to the total fidelity is 0. Therefore,

$$DP(u, f_1, f_2, \ldots, f_k) = \begin{cases} 0, \text{ if } (f_1, f_2, \ldots, f_k) \in \mathcal{F}_1 \\ -\infty, \text{ if } (f_1, f_2, \ldots, f_k) \notin \mathcal{F}_1 \end{cases} \tag{3}$$

Case 2. u is both a leaf node and a sink.

In this case, the values $f_1, \ldots f_k$ in a feasible solution must satisfy the constraints as follows:

1) $f_i \le 0$, if $u \in S_i$,
2) $0 \le f_i \le 1$, if $u \notin S_i$,
3) $\sum_{j:f_j<0} -f_j l_j \le C(u,u')$, and
4) $\sum_{j:f_j>0} f_j l_j \le C(u',u)$,

where the first constraint is because $u \in S_i$ does not need to receive the ith co-descriptor from its parent; the second constraint is to prevent u from receiving multiple copies of the same co-descriptor, which simply wastes bandwidth without any coding gain; and the third and fourth conditions are due to the edge capacity.

Let $\mathcal{F}_2$ be the set of vectors $(f_1, f_2, \ldots, f_k)$ that satisfy the above mentioned conditions. For each $f_i > 0$, u receives an i-colored co-descriptor. Therefore,

$$\begin{aligned} &DP(u, f_1, f_2, \ldots, f_k) \\ &= \begin{cases} F(\{i \mid f_i > 0 \text{ or } u \in S_i\}), \text{ if } (f_1, f_2, \ldots, f_k) \in \mathcal{F}_2 \\ -\infty, \text{ if } (f_1, f_2, \ldots, f_k) \notin \mathcal{F}_2 \end{cases} \end{aligned} \tag{4}$$

Case 3. u is an internal node and not a sink.

Because of the capacity of the edges (u',u) and (u,u'), the values $f_1, \ldots f_k$ in a feasible solution must satisfy the following constraints:

1) $f_i \le 0$ for $u \in S_i$,
2) $\sum_{j:f_j<0} -f_j l_j \le C(u,u')$, and
3) $\sum_{j:f_j>0} f_j l_j \le C(u',u)$.

Let $\mathcal{F}_3$ be the set of vectors $(f_1, f_2, \ldots, f_k)$ that satisfy the above mentioned conditions.

Let v and w be the two child nodes of u. Let f'_i and f''_i be the number of i-colored co-descriptors flowing from u to v and w, respectively. If u is not a source of color i, then the f_i paths of color i flowing into u will be splitted into v and w. That is, $f_i = f'_i + f''_i$. However, if u is a source of color i, then f'_i and f''_i can be flexiable.

Therefore, for any $(f_1, \ldots, f_k) \notin \mathcal{F}_3$, $DP(u, f_1, \ldots, f_k) = -\infty$. And for any $(f_1, \ldots, f_k) \in \mathcal{F}_3$, $DP(u, f_1, \ldots, f_k)$ can be computed by

$$\begin{aligned} &DP(u, f_1, \ldots, f_k) \\ &= \max(DP(v, f'_1, f'_2, \ldots, f'_k) + DP(w, f''_1, f''_2, \ldots, f''_k)) \end{aligned} \tag{5}$$

for all $f'_1, f'_2, \ldots, f'_k$ and $f''_1, f''_2, \ldots, f''_k$ satisfying that

$$f'_i + f''_i = f_i \text{ or } u \in S_i \tag{6}$$

Case 4. u is an internal node and sink.

First of all, $(f_1, \ldots, f_k)$ still need to satisfy the edge capacity constraint on (u',u) and (u,u'). Therefore, for a feasible solution, $(f_1, \ldots, f_k) \in \mathcal{F}_3$.

The difference between Case 4 and Case 3 is that u can now consume some of the co-descriptors and contribute to the total fidelity. If u consumes one i-colored co-descriptor and $u \notin S_i$, then $f_i = f'_i + f''_i + 1$. Therefore, (5) is changed to the following formula in Case 4:

$$\begin{aligned} &DP(u, f_1, \ldots, f_k) \\ &\quad = \max(DP(v, f'_1, f'_2, \ldots, f'_k) + DP(w, f''_1, f''_2, \ldots, f''_k) \\ &\qquad\qquad + F(\{i \mid f_i = f'_i + f''_i + 1 \text{ or } u \in S_i\})) \end{aligned} \tag{7}$$

for all $f'_1, f'_2, \ldots, f'_k$ and $f''_1, f''_2, \ldots, f''_k$ satisfying that

$$f_i - 1 \le f'_i + f''_i \le f_i \text{ or } u \in S_i \tag{8}$$

Finally, we can output $DP(root, 0, \ldots, 0)$ as the optimal total fidelity.

Note that each sink requires at most one copy of the i-th co-descriptor, $|f_i| < |T|$. Therefore, the following algorithm can be used to compute the optimal flow.

Algorithm MDC-Tree

```
1.  Traverse the tree using post-order, for each encountered node $u$
2.      for all $f_1, \ldots, f_k$ s.t. $-|T| \le f_i \le |T|$
3.          if $u$ is a leaf but not a sink
4.              use (3) to compute $DP(u, f_1, \ldots, f_k)$.
5.          else if $u$ is a leaf and a sink
6.              use (4) to compute $DP(u, f_1, \ldots, f_k)$.
7.          else if $u$ is an internal node but not a sink
8.              if $(f_1, \ldots, f_k) \notin \mathcal{F}_3$
9.                  $DP(u, f_1, \ldots, f_k) = -\infty$
10.             else use Eq-(5) to compute $DP(u, f_1, \ldots, f_k)$.
11.         else if $u$ is an internal node and a sink
12.             if $(f_1, \ldots, f_k) \notin \mathcal{F}_3$
13.                 $DP(u, f_1, \ldots, f_k) = -\infty$
14.             else use Eq-(7) to compute $DP(u, f_1, \ldots, f_k)$.
15. Output $DP(root, 0, \ldots, 0)$ as maximum total fidelity.
16. Use backtracking to compute the optimal flow.
```

Theorem 3: MDC Network Flow has a polynomial time algorithm when the network has a tree topology. The time complexity of Algorithm MDC-Tree is $O(|V| \times 2^k \times |T|^{2k})$, where T is the set of sink nodes in the flow.

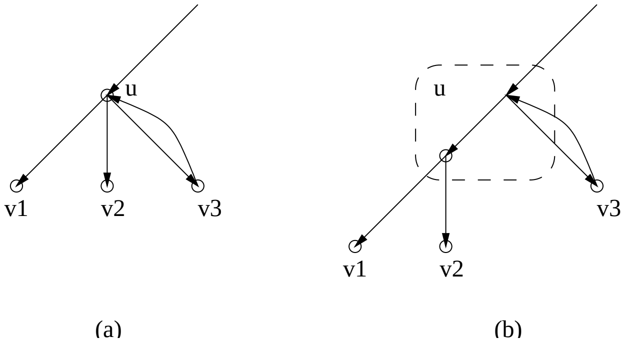
Proof: The correctness of the algorithm comes from the discussion of the four cases. We only need to analyze the time complexity. The for loop at line 2 is repeated $|T|^k$ times. Inside the for loop, the most time consuming step is line 14.

In Eq-(7), if $u \in S_i$, then there is no constraint on f'_i and f''_i. Therefore, the maximization over f'_i and f''_i can be done separately. By computing $max_{f'_i} DP(v, f'_1, f'_2, \ldots, f'_k)$ and $max_{f''_i} DP(v, f''_1, f''_2, \ldots, f''_k)$ independently, we can avoid considering the combinations of f'_i and f''_i, reducing the time complexity. As a result, the most difficult case is that u is not a source node of any co-descriptor. Then there are at most $|T|^k$ choices of $f'_1, \ldots, f'_k$. Once $f_1, \ldots, f_k$ and $f'_1, \ldots, f'_k$ are fixed, there are at most 2^k choices of $f''_1, \ldots, f''_k$ by the condition of (8). So the complexity to compute (7) is $|T|^k \times 2^k$.

For steps 2 through 14, the total complexity is $|T|^k \times |T|^k \times 2^k$. Because each of these steps is carried out for every tree node, the total complexity is $O(|V| \times 2^k \times |T|^{2k})$. ■

It is easy to see that when k, the number of different co-descriptors, is a constant, the time complexity of Algorithm MDC-Tree is polynomial.

Clearly, the same algorithm can also solve the multiple-sink rainbow network flow problem for tree network topology



(a) (b)

Fig. 3. A general tree can be reduced to a binary tree.

by letting $F(\mathcal{X}) = |\mathcal{X}|$. However, on a second reflection, a more efficient variant of the algorithm can be made possible by exploiting the property of the cost function for rainbow network flow.

Theorem 4: If $F(\mathcal{X}) = |\mathcal{X}|$, then the time complexity of Algorithm MDC-Tree can be reduced to $O(|V| \times |T|^{2k})$.

Proof: We only need to change the procedure at step 14. Because our goal is to maximize the total number of distinct co-descriptors, it is of no benefit to let a path of color i passing through u but no path of color i ending at u. So, for the computation of (7), we can change (8) to the following:

$$f_i - 1 = f_i' + f_i'' \text{ for } u \notin S_i.$$

It is easy to verify that this reduces the time complexity by a factor of 2^k. ■

For trees with degree greater than 3, we can convert it to a binary tree. Each degree k node is split into $k-2$ nodes, the edges connecting these nodes have unlimited capacity. An example is given in Fig. 3. A solution on the new binary tree can be converted to a solution on the original tree easily.

The above algorithms can be easily modified to work for undirected graphs as well.

V. DUPLICABLE MDC NETWORK FLOW

In practice, each color represents an MDC packet. For network efficiency, it is not necessary to transmit two copies of the same co-descriptor through the same edge. When a packet passing a node, the node can "duplicate" the packet and broadcast through different edges directed out of the node. Under this model, the MDC network flow problem changes slightly. The only difference is in that the capacity constraint (2) is changed to

$$\sum_{j=1}^{k} \delta(e,j) l_j \le C(e),$$

where $\delta(e,j) = 1$ if e belongs to at least one path that carries x_j, and $\delta(e,j) = 0$ if not. We call the new problem the *duplicable MDC network flow* problem.

In this section we show that when the graph has a tree topology, Algorithm MDC-Tree can be modified to optimally solve the duplicable MDC network flow problem.

The first modification is that all the f_i, f_i' and f_i'' are restricted to be -1, 0, or 1. This is because optimal flow has only one copy of the same co-descriptor to pass through an edge.

The second modification is in case 3. If $u \notin S_i$, then two things are possible: (a) color i does not reach u. Then $f_i = f_i' = f_i'' = 0$; (b) color i reaches u. Then $f_i = 1$ or $f_i' = -1$ or $f_i'' = -1$. Therefore, Eq-(6) is modified to

$$f_i = f_i' = f_i'' = 0 \text{ or } f_i = 1 \text{ or } f_i' = -1\text{or } f_i'' = -1 \text{ or } u \in S_i. \tag{9}$$

Similarly, the third modification is made to Case 4. If u receives co-descriptor i, then one of $f_i = 1$ or $f_i' = -1$ or $f_i'' = -1$ should be true. Therefore, the computation in (7) is changed to

$$\begin{aligned} &DP(u, f_1, \ldots, f_k) \\ &\quad = \max(DP(v, f_1', f_2', \ldots, f_k') + DP(w, f_1'', f_2'', \ldots, f_k'') \\ &\qquad + F(\{i \,|\, f_i = 1 \text{ or } f_i' = -1 \text{ or } f_i'' = -1 \text{ or } u \in S_i\}) \end{aligned} \tag{10}$$

for all $f_1', f_2', \ldots, f_k'$ and $f_1'', f_2'', \ldots, f_k''$ satisfying that

$$f_i = f_i' = f_i'' = 0 \text{ or } f_i = 1 \text{ or } f_i' = -1 \text{ or } f_i'' = -1 \text{ or } u \in S_i. \tag{11}$$

It is easy to verify that the same algorithm Algorithm MDC-Tree works for Duplicable MDC Network Flow after these three modifications. The time complexity is mostly determined by the computation in Eq-(10). Because of the duplicability, it does not pay to have more than one of $f_i = 1$, $f_i' = -1$, $f_i'' = -1$ to be true. Therefore, for each $u \notin S_i$, it is easy to count that the feasible choices of (f_i, f_i', f_i'') defined by Eq-(11) is 13. As a result, a careful implementation of Algorithm MDC-Tree runs in $O(|V| \times 13^k)$ time, where k is the number of different MDC co-descriptors.

VI. EXACT ALGORITHM FOR RAINBOW NETWORK FLOW

Because the rainbow network problem is NP-hard for general network topology, most likely no polynomial time algorithm exists to solve it. In order to get the optimal solution, the best one can do is to design an exponential time algorithm. Considering that there are software packages for solving linear integer programming problem of reasonable size, we formulate the problem as a 0-1 linear integer programming problem. By the following algorithm development one can solve the rainbow network flow problem exactly for a general network of modest size. Despite its high computational cost, an exact solution of the duplicable rainbow network flow problem is still valuable since it can serve as a benchmark to evaluate other heuristic but more practical algorithms.

Let $G = \langle V, E \rangle$ be the network. For each edge $e = (v_i, v_j) \in E$, let $x_{i,j}^k$ be the number of k-colored packages that pass from v_i to v_j. For rainbow network flow $x_{i,j}^k$ is either 0 or 1. In an optimal solution, one of $x_{i,j}^k$ and $x_{j,i}^k$ must be zero for passing a packet back and forth between two nodes wastes the edge capacity without any coding gain. This yields the following constraint:

$$x_{i,j}^k + x_{j,i}^k \le 1$$

In a special case we let $x_{i,i}^k = 0$ as convention.

For each node v_j, we use a binary variable y_j^k to indicate whether v_j gets a copy of k-colored packet in the flow. For v_j that is a source of the color k, $y_j^k = 1$ automatically. For v_j that is not a source of the color k, obviously

$$y_j^k \le \sum_{i=1}^{n} x_{i,j}^k.$$

Now consider the situation when $x_{i,j}^k = 1$. We assume that any network node can duplicate and forward a packet. In this case v_i has to first possess a copy of k-colored packet before transmitting duplicated copies to different destinations, if required. The capability of duplicating a packet by a network node leads to the constraint:

$$\forall j \quad x_{i,j}^k \leq y_i^k. \tag{12}$$

The edge capacity constraint is converted to one of the following two formulae, depending on whether the network graph is directed or not. For directed graph, let $c_{i,j}$ be the capacity from v_i to v_j, and here $c_{i,j}$ may not equal to $c_{j,i}$. The edge capacity constraint is:

$$\sum_k x_{i,j}^k \leq c_{i,j} \tag{13}$$

For undirected graph, let $c_{i,j}$ be the capacity of edge (v_i, v_j), and we have $c_{i,j} = c_{j,i}$. The edge capacity constraint becomes:

$$\sum_k (x_{i,j}^k + x_{j,i}^k) \leq c_{i,j} \tag{14}$$

For each sink $v_j \in T$, the fidelity function is $\sum_k y_j^k$. Therefore, the total fidelity is

$$\sum_{v_j \in T} \sum_k y_j^k$$

Then the duplicable rainbow network flow problem can be cast into the following linear integer programming problem:

$$\begin{aligned}
\max \quad & \sum_{v_j \in T} \sum_k y_j^k && (15)\\
\text{subject to} \quad x_{i,j}^k + x_{j,i}^k &\leq 1 \\
y_j^k &\leq \sum_i x_{i,j}^k \quad \text{if } v_j \notin S_k \\
y_j^k &= 1 \quad \text{if } v_j \in S_k \\
x_{i,j}^k &\leq y_i^k \\
\sum_k (x_{i,j}^k + x_{j,i}^k) &\leq c_{i,j} && (16)\\
y_j^k = 0 \text{ or } 1, \quad & x_{i,j}^k = 0 \text{ or } 1 && (17)
\end{aligned}$$

The constraint Eq-(16) shall be replaced by Eq-(13) if the network graph is directed.

For non-duplicable version of the rainbow network flow problem the above development is still valid, if one changes the constraint Eq-(12) to

$$\sum_j x_{i,j}^k \leq y_i^k. \tag{18}$$

We applied a commercial integer programming solver Cplex to the proposed integer programming problem of rainbow network flow. The solutions found by Cplex were on average 15% better than a greedy broadcast algorithm. Our experiments were conducted on simulated networks of 120 nodes, for four colors, maximum edge capacity of 3, and an average node degree of three. Solving a problem instance of this size took only few minutes.

By modifying Eq-(15) to the total fidelity function of Eq-(1), the same formulation can be used to solve the general MDC network flow problem. However, the underlying integer programming problem is not linear anymore, and the solution becomes much harder.

Similarly, the duplicable MDC network flow problem can be converted into one of integer programming.

VII. CONCLUSION

We introduced the MDC network flow problem of optimizing the flows of MDC packets for maximum overall reconstruction quality of all receivers. We proved that MDC network flow problem is Max-SNP hard. As a result, the problem is unlikely to have a polynomial time approximation scheme that approximates the optimal solution arbitrarily well. We then proposed polynomial time algorithm for a practical case of the MDC network flow problem, where the network topology is a tree and the number of descriptors is limited. The Duplicable MDC network flow problem is also studied. At last, a practical algorithm, which runs potentially in exponential time, is given to find the optimal solution of the rainbow network flow problem.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Info. Th.*, vol. 46, no. 4, pp. 1204-1216, 2000.

[2] G. Al-Regib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively compressed 3-d meshes," in *Proc. of IEEE INFOCOM*, 2002.

[3] R. Arean, J. Kovacevic, and V. Goyal, "Multiple description perceptual audio coding with correlation transforms," *IEEE Trans. Speech Audio Proc.*, vol. 8, no. 2, 2000.

[4] V. Chande and N. Farvardin, "Joint source-channel coding for progressive transmission of embedded source coders," in *Proc. DCC'99 Data Compression Conference*, 1999, pp. 52–61.

[5] P. Chou, Y. Wu, and K. Jain, "Network coding for the internet," in *IEEE Communication Theory Workshop*, 2004.

[6] S. Dumitrescu and X. Wu, "Optimal two-description scalar quantizer design," *Algorithmica*, vol. 41, no. 4, pp. 269–287, 2005.

[7] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error protected packetization of scalable code streams," *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp. 230–239, 2004.

[8] M. Fleming, Q. Zhao, and M. Effros, "Network vector quantization," *IEEE Trans. Inform. Th.*, to appear.

[9] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, 2001.

[10] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proc. Information Processing in Sensor Networks 2004*, 2004.

[11] W. Lee, M. Pickering, M. Frater, and J. Arnold, "A robust codec for transmission of very low bit-rate video over channels with bursty errors," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 10, 2000.

[12] S. Miller, "Fail-safe transmission without standby facilities," *Bell Labs. Tech. Rep. TM80-136-2*, 1980.

[13] A. E. Mohr, R. E. Ladner, and E. Riskin, "Approximately optimal assignment for unequal loss protection," in *ICIP 2000 International Conf. on Image Processing*, vol. 1, 2000.

[14] D. Muresan and M. Effros, "Quantization as histogram segmentation: globally optimal scalar quantizer design in network systems," in *Proc. Data Compression Conf.'2002*, 2002.

[15] C. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *Journal Computer and System Sciences*, vol. 43, pp. 425–440, 1991.

[16] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 1, 1999.

[17] P. Sherwood and K. Zeger, "Progressive image coding on noisy channels," in *Proc. DCC'97 Data Compression Conference*, 1997, pp. 72–79.

[18] V. A. Vaishampayan, "Design of multiple-description scalar quantizers," *IEEE Trans. Inform. Th.*, vol. 39, no. 3, pp. 821-834, 1993.

[19] V. A. Vaishampayan, N. Sloane, and S. Servetto, "Multiple description vector quantization with lattice codebooks: Design and analysis," *IEEE Trans. Inform. Th.*, vol. 47, no. 5, pp. 1718-1734, 2001.

[20] X. Wu, B. Ma, and N. Sarshar, "Rainbow network problems and multiple description coding," in *2005 IEEE Symp. Info. Th.*, 2005, submitted.