Rainbow Network Flow of Multiple Description Codes

Xiaolin Wu[1], Bin Ma[2], Nima Sarshar[1]
[1] Dept. of Electrical & Computer Engineering, McMaster Univ., Hamilton, ON, Canada
[2] Dept. of Computer Science, Univ. of Western Ontario, London, ON, Canada

Abstract

This paper is an enquiry into the interaction between multiple description coding (MDC) and network routing. We are mainly concerned with rate-distortion optimized network flow of an MD source from multiple servers to multiple sinks. Maximizing a collective metric of the MD source reconstructed at all sinks, constrained by edge capacities, is a very different problem from conventional maximum network flow. The objective function involves not only the flow volume but also the diversity of the flow contents (distinct MDC descriptions), hence the term of rainbow network flow (RNF). The RNF problem is also closely related to lossy network coding. For general network topology, general fidelity function, and an arbitrary distribution of MDC descriptions on the servers, we prove that the RNF problem is Max-SNP-hard, i.e., there is no polynomial-time algorithm to even approximate the optimal solution with arbitrarily good approximation ratio, unless P=NP. However, the problem becomes tractable in many practical scenarios, such as when MDC is balanced with descriptions of the same length and importance, when all source nodes have the complete set of MDC descriptions, and when the network topology is a tree or has only one sink. Polynomial-time RNF algorithms are developed for these cases.

Keywords: Multiple description coding, network routing, network coding, optimization, complexity.

I. Introduction

Packet switched lossy networks, such as the Internet, peer-to-peer, ad hoc, and diversity wireless networks, inevitably experience packet losses and delays problems. 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 design of network-aware codes. The need for more sophisticated codes to compensate for less network provisions has led to a proliferation of research literature on multiple description coding (MDC) for packet-switched networks and erasure channels [4].

MDC is an effective technique to maintain the quality of service at times of network congestions and server breakdowns by offering a client multiple accesses to a given content. Different descriptions of an MDC-coded source can be transmitted to the client via different paths and from different servers in the network. In this paper

Parts of the paper were presented at NetCod 2005 and ISIT 2005. The work is supported by Natural Sciences and Engineering Research Council of Canada.

we consider a network of a fixed topology, modeled by a graph $G = \langle V, E\rangle$. The nodes in the set V correspond to servers, clients, and possibly relays in the network. The edges in the set E represent directed links between these nodes. Each edge has a capacity, reflecting the capacity of the underlying channel.

Suppose that a source is coded by an MDC into a set of $N \geq 2$ descriptions $X = \{x_1, x_2, \cdots, x_N\}$. Let $\mathcal{X}_k = \{x_{k,1}, x_{k,2}, \cdots, x_{k,n_k}\} \subseteq X$ be the subset of descriptions at server k. Subsets $\mathcal{X}_k$ and $\mathcal{X}_j$ residing 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 in case of server or link failures. Any client in the network can decode the MDC-coded source upon receiving a subset $\mathcal{X} \subseteq X$ of descriptions with fidelity $F(\mathcal{X})$. The quality of service (QoS) for all the network clients depends on the network graph $G = \langle V, E\rangle$, the edge capacities, the distribution of the descriptions at the servers, and most of all on the network flow of the MDC descriptions.

An interesting and important problem arises naturally when considering how to best route MDC descriptions in a network. Given the topology and edge capacities of a network and given the distribution of MDC descriptions on a set of servers, what are the optimal network flows of the different descriptions in terms of maximizing the reconstruction quality at one or a group of sinks (clients)? The optimal network flow of MDC descriptions in rate-distortion sense behaves very differently from maximum flow of commodities in network problems treated in classic operation research literature. Computationally, the former is far more complex than the latter because the reconstruction fidelity achieved by a decoder is not additive of the fidelities offered by the individual descriptions received. Our objective is to maximize not the total volume of commodities (descriptions) that can flow into the sink(s), but rather the total fidelity achievable over all possible *sets of descriptions* that can flow into the sink(s), constrained by edge capacities. In the case of balanced MDC of equally important packets of the same size, the coding gain realized by a sink becomes the number of *distinct* packets received by it. For a given sink receiving two or more copies of the same MDC packet makes no additional coding gain. For this unique property of MDC, we call the problem of optimizing the routing of MDC code streams the rainbow network flow (RNF) problem. The terminology carried an intuitive connotation: distinctively color the MDC descriptions and optimize the network flows to achieve the rainbow effect of getting as wide a spectrum of colors as possible at the sinks.

The RNF problem is also an integral aspect of joint network-source coding (JNSC), a study recently initiated by the authors [13]. The inquiry of JNSC is about how to code a (usually real valued) source, communicate and reconstruct the coded source in a network to a maximal collective fidelity over a given set of sinks, while the flows of the code streams meet the edge capacities of the network. JNSC can be considered as a lossy version of the

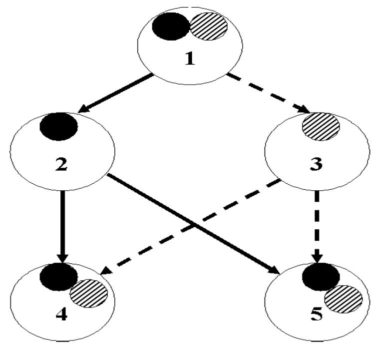

Fig. 1. An example of flow of a two description code.

(lossless) network coding problem [1], since the reconstruction dose not need to be perfect. An interesting and profound observation made in our study of JNSC is that MDC, if coupled with optimized network routing, can improve the overall rate-distortion performance. Remarkably, this is true *even when all the communication links of the network are error free*, in spite of a prevailing folklore that the redundancy of MDC can only be justified in the presence of channel errors (indeed, the research of MDC seemed solely motivated by the desire for robust networked communications against packet erasure errors). Our point can be illustrated by the following example.

In Fig. 1, a server (node 1) feeds a coded source into a network of four sink nodes (nodes 2-5). The goal is to have the best reconstruction of the source at each of these four nodes. All link capacities are C bits per source symbol. MDC encodes the source into two descriptions (shown by solid and dashed boxes in the figure), each of rate C. Descriptions 1 and 2 are sent to nodes 2 and 3 respectively. Node 2 duplicates description 1 and sends to nodes 4 and 5 each a copy, while node 3 duplicates description 2 and sends a copy to nodes 4 and 5. The RNF consists six directed paths: $1 \to 2$, $1 \to 2 \to 4$, $1 \to 2 \to 5$, $1 \to 3$, $1 \to 3 \to 4$, $1 \to 3 \to 5$. To distinguish these paths by color (description), we form two sets of colored paths: $P_1 = \{1 \to 2, 1 \to 2 \to 4, 1 \to 2 \to 5\}$ and $P_2 = \{1 \to 3, 1 \to 3 \to 4, 1 \to 3 \to 5\}$

To see how the nodes in the network benefit from MDC, let $D_1(C), D_2(C), D_{12}(C)$ be the distortion in reconstructing the source given description 1 or 2 or both. Lets minimize the average distortion at all nodes 2 through 5 in Fig. 1 (assuming all sink nodes with equal weight)

$$\bar{d} = \frac{2D_{12}(C) + D_1(C) + D_2(C)}{4} \tag{1}$$

For an iid Gaussian source of unit variance, the achievable distortions of two description coding are completely derived by Ozarow [10]. The symmetry in indices 1 and 2 ensures that (1) is minimized when the two descriptions are balanced, that is, $D_1(C) = D_2(C) = D$. Ozarow's result, when specialized to balanced MDC states that the following set of distortions are achievable:

$$\begin{aligned} D_1 &= D_2 = D \geq 2^{-2C} \\ D_{12} &\geq \frac{2^{-4C}}{(D + \sqrt{D^2 - 2^{-4C}})(2 - D - \sqrt{D^2 - 2^{-4C}})} \end{aligned} \tag{2}$$

The minimum value $\overline{d}^*_M(C)$ of the average distortion (1) under the constraints of (2) can be computed via a convex optimization procedure, because the region (2) is convex. On the other hand, by separating source from network coding, the reconstruction distortion at nodes 2 through 5 (and hence the average distortion over all these nodes) is at best $d_S(C) = 2^{-2C}$. One can easily verify that $\overline{d}^*_M(C) < d_S(C)$ for all $C > 0$. In other words, for all $C > 0$ there exists a balanced two description code for which the average distortion over all sink nodes is strictly less than the average distortion achievable by any separate source and network coding scheme.

The inefficiency of separate source and network coding lies in that even though nodes $4, 5$ have twice the incoming capacity compared to nodes $2, 3$, their reconstruction error ($d_4 = d_5$) is bounded by the reconstruction error of the weaker nodes ($d_2 = d_3$). Unlike lossless coding, lossy codes can play a tradeoff between the reconstruction errors at different nodes, generating a much larger set of achievable distortion 4-tuples (d_2, d_3, d_4, d_5). These tradeoffs are essential in practice. For instance, in networked multimedia applications over the Internet, where the network consists of a set of heterogenous nodes, the experience of a user with broadband connection should not be bounded by that of a user with a lesser bandwidth.

Extending the above example, we showed that given a rainbow network flow of MDC descriptions, optimal MDC design can be posed and solved as a convex optimization problem [13]. If reciprocally the RNF problem can be solved satisfactorily, then a practical JNSC approach will be to optimize the MDC design and the MDC routing in turn, with one of the two fixed at a time. The potential advantages of jointly optimizing the design and the routing of MDC provide another motivate for studying the RNF problem.

The main contributions of this paper are listed and presented as follows. In Section II we formulate the rainbow network flow problem. Then we present complexity results of the problem in Section III. After proving the RNF problem to be MAX-SNP-hard, we turn our attention to some practically important cases for which polynomial-time algorithms exist. Section IV develops an algorithm to solve the optimal RNF problem with respect to a single sink. Section V derives a dynamic programming algorithm that can solve the optimal RNF problem for an arbitrary distribution of MDC descriptions if the network has a tree topology. This algorithm is then generalized in Section VI to the multicast rainbow network flow problem in trees, where nodes are allowed to multicast a received description. Section VII presents a family of algorithms for the case of so-called full source spectrum, where all servers can supply all descriptions of an MDC-coded source. Section VIII concludes the paper with suggestions of future research problems.

II. PROBLEM FORMULATION AND NOTATIONS

The problem of *Rainbow Network Flow* (RNF) of MDC is posed in the following setting:

- 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 descriptions of rates $R_1, R_2, \cdots, R_N$;
- a set of server nodes $S \subset V$.
- each server node $s_i \in S$ has a set of descriptions $\kappa_1, \kappa_2, \ldots, \kappa_{N_i}$, called the spectrum $\Psi_S(s_i)$.
- a set of sink nodes $T \subset V$.

For convenience we say that description n has color n. A colored network flow consists of N sets (some may be empty) of directed flow paths, $P_1, \ldots, P_N$. A flow path $p \in P_n$, P_n being a non-empty set, carries a copy of description n from a server node $s \in S$ such that $n \in \Psi_S(s)$ to a sink node $t \in T$. Let $\wp_E(e)$ be the set of flow paths that pass through edge $e \in E$, and $\wp_T(t)$ be the set of flow paths that end at the sink node $t \in T$. Denote by $\kappa(p)$ the color of path p, i.e., $\kappa(p) = n$ if $p \in P_n$. Define the spectrum of an edge $e \in E$ to be the color set

$$\Psi_E(e) \equiv \bigcup_{p \in \wp_E(e)} \{\kappa(p)\}; \tag{3}$$

likewise, define the spectrum of a sink $v \in T$ to be the color set

$$\Psi_T(t) \equiv \bigcup_{p \in \wp_T(t)} \{\kappa(p)\}. \tag{4}$$

Let $\mathcal{F}(\Psi)$ be the reconstruction fidelity achieved by decoding a subset $\Psi \subseteq \{1, 2, \cdots, N\}$ of the N MDC descriptions. The objective of RNF is to find a colored network flow to maximize a weighted fidelity measure over all the sink nodes, while satisfying all the edge capacities, or stated as the following constrained optimization problem:

$$\max_{P_1, \ldots, P_N} \sum_{t \in T} w(t) F(\Psi_T(t)) \tag{5}$$

subject to

$$\sum_{p \in \Psi_E(e)} R_{\kappa(p)} \leq C(e), \quad \forall e \in E. \tag{6}$$

with $w(t)$ being a weighting function to prioritize different sinks in optimizing MDC network flows (by user fees, urgency, and etc.)

An important nuance of the inequality (6) is that it allows for duplication (i.e., multicast) of a description by relay nodes. In other words, two or more flow paths of color i that pass through an edge $e \in E$ consume a bandwidth of only R_i. This corresponds to network routers that can duplicate a received data stream and send the copies through multiple links. If the network nodes cannot duplicate any incoming description, then the edge capacity constraint of (6) should be changed to

$$\sum_{p \in \wp_E(e)} R_{\kappa(p)} \leq C(e), \quad \forall e \in E \tag{7}$$

We refer to the variant of (6) as the multicast rainbow network flow problem, symbolized by RNF* in the sequel, to distinguish it from the second variant where relay nodes are pure switches. Duplicating MDC descriptions is a simple and yet powerful operation of lossy network coding that can greatly improve the utilization of network capacities, as demonstrated in the examples of Figs. 1, 2, and analyzed in [13]. For this reason RNF* is rate-distortion more efficient than RNF.

The meanings of the above definitions and notations may be better conveyed by visualizing an example. Fig. 2 depicts the optimal routing of an MDC of two descriptions (descriptions 1 and 2 are labeled by solid and dashed icons and arrows, respectively) in a network of six nodes. The server nodes are node 2 that has both descriptions and node 4 that has only description 1. Nodes 1, 3, 5 are sinks, and nodes 0 and 1 are relays (routers). All edges have a capacity of 1 except for the edge $(0, 5)$ of capacity 2. The diagram on the left presents an optimal solution of RNF*, whereas the diagram on the right shows an optimal solution of RNF. Comparing the left and right diagrams we see the difference between RNF* and RNF. In the case of RNF*, the relay node 0 duplicates description 1 and sends a copy to each of sink nodes 3 and 5. But because node 0 cannot duplicate any received description in RNF, sink nodes 3 and 5 cannot both get all the two descriptions as in RNF*. In this example RNF wastes the capacity of edge $(0, 3)$, achieving lower fidelity at sink node 3.

To be familiarized with the notations, please refer to Table II for the notation instantiation in the optimal RNF* solution depicted in Fig. 2.

Also, we define the *Undirected Rainbow Network Flow Problem* (URNF) the same as the RNF problem except that the edges and paths in the definition are undirected.

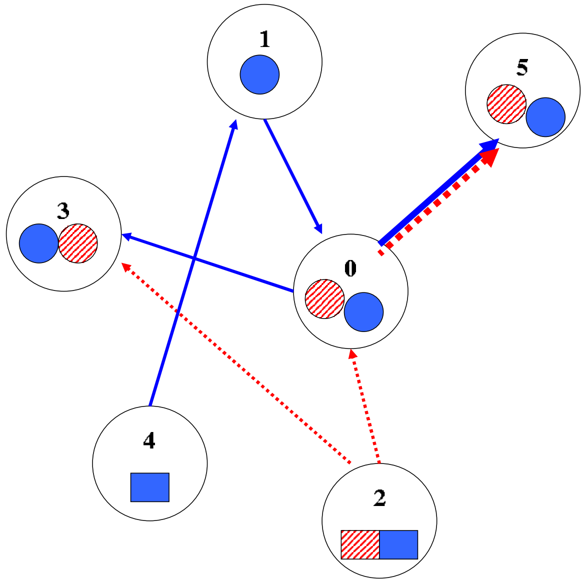

Fig. 2. Optimal routing of an MDC of two descriptions, marked by solid and dashed icons and edges, for RNF* (left) and RNF (right). All edges have capacity 1 except for edge $(0,5)$ of capacity 2.

Notation instantiation in optimal RNF* solution for the above example

Problem Input:
$N = 2$ (two MDC descriptions)
$R_1 = R_2 = 1$
$C(e) = 1$ for all $e \in E$ other than $C(0,5) = 2$
$S = \{2,4\}$, $\Psi_S(1) = \{2,4\}$, $\Psi_S(2) = \{2\}$
$T = \{1,3,5\}$

RNF^* **Output:**
Optimal sets of colored flow paths:
$P_1 = \{4 \to 1, 4 \to 1 \to 0 \to 3, 4 \to 1 \to 0 \to 5\}$
$P_2 = \{2 \to 0 \to 5, 2 \to 3\}$

$\wp_E(4,1) = \{4 \to 1, 4 \to 1 \to 0 \to 3, 4 \to 1 \to 0 \to 5\}$
$\wp_E(1,0) = \{4 \to 1 \to 0 \to 5, 4 \to 1 \to 0 \to 3\}$
$\wp_E(0,3) = \{4 \to 1 \to 0 \to 3\}$
$\wp_E(2,0) = \{2 \to 0 \to 5\}$
$\wp_E(2,3) = \{2 \to 3\}$
$\wp_E(0,5) = \{4 \to 1 \to 0 \to 5, 2 \to 0 \to 5\}$

$\wp_T(1) = \{4 \to 1\}$
$\wp_T(3) = \{4 \to 1 \to 0 \to 3, 2 \to 3\}$
$\wp_T(5) = \{4 \to 1 \to 0 \to 5, 2 \to 0 \to 5\}$

$\Psi_E(4,1) = \Psi_E(1,0) = \Psi_E(0,3) = \{1\}$
$\Psi_E(2,3) = \Psi_E(2,0) = \{2\}$
$\Psi_E(0,5) = \{1,2\}$

$\Psi_T(1) = \{1\}$
$\Psi_T(3) = \Psi_T(5) = \{1,2\}$

TABLE I

NOTATION INSTANTIATION IN THE OPTIMAL RNF* SOLUTION DEPICTED IN FIG. 2.

The RNF problem definition is for general fidelity function $F(\Psi)$, in which $F(\Psi)$ accounts for not only the correlations between MDC descriptions such as those of multiple description quantizers, but also the sequential decoding dependency between descriptions such as those in layered multi-resolution source coding.

On surface the RNF problem might appear similar to the multicommodity network flow problem, by viewing different MDC descriptions as different commodities to be transported in a network. However, the two problems are very different due to the unique fidelity (distortion) metric of MDC. In RNF a sink does not demand any particular commodity or commodities, rather it desires to have as many *distinct* descriptions as possible (i.e., which commodities do not matter so long they are unique). For the same reason the RNF problem also differs from conventional network flow of a single commodity. Maximizing the total flow volume into the sinks, which is the objective of the latter problem, is not optimal since the conventional maximum flow may carry duplicates of an MDC description to a sink, occupying edge capacities for no coding gains.

III. Complexity Results of RNF Problems

This section presents the main complexity results for RNF with the corresponding proofs placed in the appendices. We start with the complexity of the RNF problem without duplication. First, we prove that the RNF problem is NP-hard by showing that even a special case of RNF problem is NP-hard. The special case, which is called Cardinality Rainbow Network Flow (CRNF), is when $F(\Psi) = |\Psi|$ and $w(t) = 1$ for all $t \in T$ in (5).

Theorem 1: Cardinality rainbow network flow problem is NP-hard, even for directed acyclic planar graphs.
The proof of this theorem is in Appendix A.

It is easy to see that the reduction used in Appendix A also works when the graph is undirected. Therefore, we have

Theorem 2: Undirected cardinality rainbow network flow problem is NP-hard even for undirected planar graphs.

The CRNF problem, although being only a special case of RNF, is important in practice. The popular technique of uneven error protection (UEP) of scalable source code stream using Reed-Solomon code makes all MDC descriptions (packets) to have the same size and same importance. Consequently, the fidelity function $F(\Psi)$ only depends on the cardinality of Ψ not the particular composition of Ψ. Maximizing reconstruction fidelity is equivalent to maximizing the number of distinct descriptions received by all sinks. Appendix A proves that CRNF problem is NP-hard if the network has multiple sinks and multiple servers. However, if the optimization is with respect to a single sink, the

CRNF problem can be reduced to one of classic maximum network flow and hence is polynomially solvable. The algorithm for CRNF of a single sink will be given in the next section.

For general MDC the reconstruction fidelity $F(\Psi)$ does depend on the composition of Ψ to account for different coding gains made by different combinations of MDC descriptions, and also for arbitrary dependencies in decoding these descriptions. The increased combinatorial complexity as opposed to the set cardinality suggests that the general RNF problem ought to be "harder" than the CRNF problem. Indeed, we can prove that the RNF problem remains NP-hard even the network has only one sink node. In fact, we will prove a stronger complexity result that the single-sink RNF problem is Max-SNP-hard. Namely, there is a constant $\epsilon > 0$, such that no polynomial-time algorithm can approximate RNF problem with ratio better than $1 + \epsilon$, unless P=NP.

Theorem 3: The rainbow network flow problem is Max-SNP-hard even if there is only one sink node, all MDC descriptions have equal rate, and the network topology is a tree.

The proof of the theorem is given in Appendix B.

The reduction used in Appendix B also applies to undirected graphs. Thus, we conclude

Theorem 4: Undirected rainbow network flow problem is Max-SNP-hard even if there is only one sink node, all MDC descriptions have equal rate, and the network topology is a tree.

It turns out that the reductions in the proofs of the above four theorems also work when the nodes can duplicate MDC descriptions. Consequently, we have the following theorem:

Theorem 5: Theorem 1 and 2 hold for CRNF*. Theorem 3 and 4 hold for RNF*.

The proof of Theorem 5 is given in Appendix C.

Although RNF problem is NP-hard in general, there are practically important cases for which polynomial-time algorithms exist. We will examine some of these cases in the next four sections.

IV. Optimal RNF for Single Sink

The rainbow network flow problem can be greatly simplified if the flow of MDC descriptions from diversity servers is optimized with respect to a given sink. This is particularly so for RNF* since the optimal solution for a single sink does not require duplication of any MDC description.

First, let us reexamine the CRNF problem for a single sink, although optimal CRNF was proven in the previous section to be NP-hard for multiple sinks. By the definition of CRNF, all MDC descriptions have the same importance

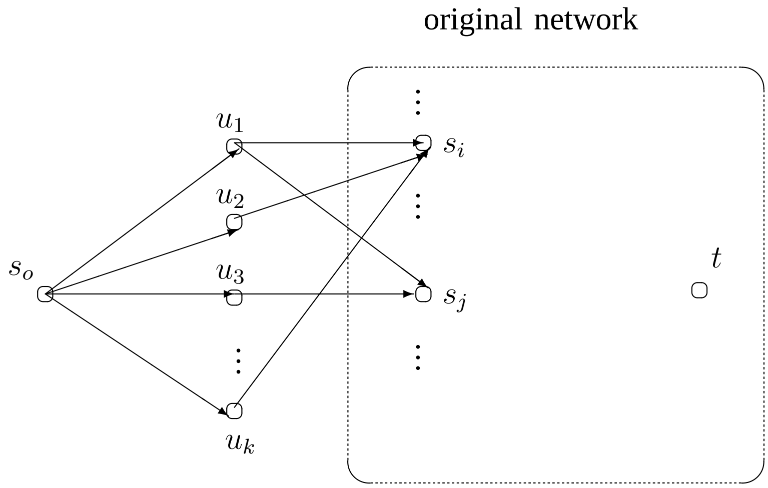

Fig. 3. Expanded graph with added edges from u_i to the servers that have color i.

and the same rate R, which is treated as the unit rate of 1 for convenience. To the only sink $t \in T$, the optimal MDC network flow that achieves the maximum fidelity at the sink t will necessarily have K flow paths of K distinct colors ($K \le N$), or there is at most one path with respect to each color. These observations allow us to reduce the single-sink CRNF problem to one of conventional maximum network flow, which we call maximum monochrome network flow to distinguish it from maximum-fidelity rainbow network flow.

For each color i, we add a new vertex u_i to the graph G, and add an edge (u_i, s) for each $s \in S$ such that $i \in \Psi_S(s)$. The capacity of edge (u_i, s) is set to 1. Then we add a "super server node" s_o, and edges (s_o, u_i) with capacity 1 for each i. Fig. 3 depicts the resulting expanded graph. We assume that the capacities of all edges in the network are integers. The above construction is valid for arbitrary network topology, and it equates the optimal solution for the single-sink CRNF problem to the maximum monochrome flow from s_o to t in the expanded graph. The latter problem can be solved easily by Goldberg and Tarjan's maximum flow algorithm in $O(|E||V|\log(|V|^2/|E|)$ time [3]. [1]

The resulting maximum monochrome flow of volume K corresponds to K paths $p_1, \ldots, p_K$ from s_o to t. And each edge e of the graph appears in at most $C(e)$ of the K flow paths. Since the K edges (s_o, u_i), $i = 1, \ldots, K$, are the only outgoing edges of s_o, and each of them has capacity 1, the resulting K flow paths must each go through a different u_i. Referring to Fig. 3, if a path p_i from s_o to t first reaches node u_{k_i}, then it must immediately enter a node $s \in S$. Denote by p'_i the remainder subpath of p_i from the node s to t. Clearly, the subpath p'_i is a path from the server s to the sink t of color k_i for the original CRNF problem. Therefore, the set of paths $\{p'_i\}_{i=1,\ldots,K}$,

[1]It is possible to solve it faster in $O(|E||V| + |V|^{2+\epsilon})$ time by a more complicated algorithm [7].

constitutes a solution for the original single-sink CRNF problem. The procedure described above is summarized by the pseudo-code *SingleSink* below.

Algorithm SingleSink
Input $G = \langle V, E \rangle$; server set S with spectrum $\Psi_S(s)$, $s \in S$; sink $t \in V$;
Output K paths p'_i, $1 \leq i \leq K \leq N$, each with a different color.
1. For each color i,
   add a new vertex u_i to V;
   for each $s \in S$ such that $i \in \Psi_S(s)$
     add an edge (u_i, s) with $C(u_i, s) = 1$ to E.
2. Add a new vertex s_o to V.
3. For each color i,
   add an edge (s_o, u_i) with capacity $C(s_o, u_i) = 1$ to E.
4. Compute the maximum flow from s_o to t in the new graph, resulting in $p_1, \ldots, p_K$, the K different paths from s_o to t that compose the flow.
5. For i from 1 to K,
   remove the first two edges in p_i to get a new path p'_i.
6. Output $p'_1, \ldots, p'_K$.

Theorem 6: The flow computed by Algorithm SingleSink is an optimal solution of CRNF.

Proof: Clearly, the paths computed by the algorithm form a feasible solution of CRNF. We only need to show that the solution is optimal. This can be proved by contradiction.

Suppose that we have another solution of $K' > K$ paths with different colors $i_1, \ldots, i_{K'}$. Let the K' paths be $q'_1, \ldots, q'_{K'}$. Let q'_i have color k_i and q'_i connect from a server node $s_i \in S$ such that $k_i \in \Psi_S(s_i)$ to the sink t. Then there is an edge (u_{k_i}, s_i) in the expanded graph. As a result, $q_i = (s_o, u_{k_i}) \cdot (u_{k_i}, s_i) \cdot q'_i$ is a path from s_o to t in the expanded graph. The collection of these paths $\{q_i\}_{i=1,\ldots,K'}$ forms a monochrome network flow from s_o to t whose total flow volume is $K' > K$. This is contradictory to the optimality of step 4. ■

For CRNF on an undirected graph with single sink, the above algorithm works as well with virtually no modification. The only concern is that the constructed graph (Fig. 3) now has all the newly added edges directed, and all the edges in the original network undirected. However, in the traditional network flow problem, each undirected edge can be converted to two opposite directed edges with the same capacity. The optimal solution of the resulting directed graph will be the same as the optimal solution of the original graph with undirected edges. As a result, Algorithm SingleSink is also an efficient algorithm for CRNF on undirected graphs with single sinks. We have the following theorem.

Theorem 7: Algorithm SingleSink computes an optimal solution of the single-sink undirected CRNF problem.

Algorithm SingleSink can be modified to provide practical solution to general RNF problems with single sink as well. For unbalanced MDC where $R_i \neq R_j$, $i \neq j$, we can change all edge capacities $C(s, u_i)$ and $C(u_i, v)$, where $v \in S_i$, from 1 to R_i, $1 \leq i \leq N$ (referring to Fig. 3). Then Algorithm SingleSink can maximize the total number of bits flowing into sink t from all server nodes. The topology of the expanded graph in Fig. 3 ensures that no parts can be duplicated from the same MDC description. Whether the algorithm achieves the maximum fidelity possible allowed by the network capacity, like in the case of CRNF, depends on the design of MDC codes and the functionality of the network.

Consider the MDC codes for which the fidelity $F(\Psi)$ achieved by decoding the set Ψ of descriptions is monotone in the total rate of all descriptions in Ψ, namely,

$$F(\Psi_1) \geq F(\Psi_2) \quad \text{if} \sum_{\kappa \in \Psi_1} R_\kappa \geq \sum_{\kappa \in \Psi_2} R_\kappa. \tag{8}$$

Many MDC codes satisfy the rate monotonicity, including those generated by the priority encoding transmission (PET) technique and many codes of multiple description quantization. Suppose that the network allows a description to be split into multiple paths in transmission. Then the solution found by Algorithm SingleSink is optimal provided that a description is either received in its entirety, or not at all. The optimality cannot be guaranteed if fractional description is received because the fidelity function F is defined on a set of complete descriptions.

A sure way of receiving a description in full is not to allow the description to be split and sent via different paths from s to t. Unfortunately, adding this constraint converts the underlying optimization problem to be the known NP-hard problem of unsplittable maximum network flow [8].

The above discussions lead to a new, practically important, and challenging MDC flow problem: maximizing the fidelity at the sink(s) with splitable flow of MDC descriptions under the constraint that all of the received MDC descriptions are complete. This constraint is more forgiving than that of the unsplittable network flow problem in that a description can be split during transmission as long as it arrives at a sink in whole. The relaxation is desirable because the network edge capacities can be better utilized if a large description can be sent in parts and assembled at a sink. This also agrees with the actual mechanism of packet switched networks, where a large description can be broken into multiple data packets that may be routed differently. The issue of splittable flow will be addressed again in Section VII.

V. Optimal RNF for Tree Topology

If the network topology is a tree and the number N of MDC descriptions is a constant independent of the network size (typically true in the practice of MDC coding), then the optimal RNF problem can be solved in polynomial time. The tree topology is common in local area networks (LAN) and in sensor networks [5].

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 convert it to 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 the new graph G' is a tree, we say that G has a tree topology. Fig. 4(a) shows an example of a directed graph with a tree topology.

We develop an RNF algorithm first for unrooted binary trees, and then extend it to arbitrary unrooted trees. We can make G a binary tree rooted at an arbitrary leaf node $v \in V$ by inserting a dummy vertex v^* and a dummy edge (v, v^*) into G. Fig. 4 illustrates the tree construction on the directed graph G. Consequently, each vertex $u \in V$ is either an internal node having two child nodes, or a leaf node of no descendants. The subtree rooted at a node $u \in V$ can also be naturally defined. Note that the dummy vertex is purely for the clarity of the algorithm, and it has no effect on the optimal solution for the original tree. As a result, this dummy vertex can be added at an arbitrary leaf.

In this section we assume that the relay nodes of the network cannot duplicate any received color. In the next section we will modify the optimal RNF algorithm for networks of tree topology in which the relay nodes can duplicate colors.

For each internal node u, let u' be the parent node of u. If there are x_j paths of color j passing through u and u' in the optimal flow, then those x_j paths have to go through u and u' in the same direction. This is because passing a color back and forth through an edge would consume the edge capacity without increasing the number of distinct colors to arrive at the sinks. Therefore, we can assign a unique signed flow volume f_j to the edge connecting u' and u: $f_j = x_j \geq 0$ if the paths flow from u' to u, and $f_j = -x_j \leq 0$ otherwise. This way we can unambiguously say that there are f_j paths flowing into u from its parent u'.

We take a dynamic programming (DP) approach to solving the problem. Let f_i be integers. Let $DP(u, f_1, f_2, \ldots, f_k)$ be the maximum total fidelity of the sinks in the subtree rooted at u, given that f_j paths of color j flow into u from its parent. Let v, w be the two children of u. The dynamic programming algorithm relies on a recurrence relation to compute $DP(u, f_1, f_2, \ldots, f_N)$ from $DP(v, f'_1, f'_2, \ldots, f'_N)$ and $DP(w, f''_1, f''_2, \ldots, f''_N)$. To derive the recurrence we need to differentiate the following four cases, depending on the position and function of a network node u: whether u is a sink, a server, a relay (router), or some combination of the above.

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

In a practical network the node u is a pure server ($u \in S$) since it is pointless to place a router at a terminal node. We ban any description to flow into u, or $f_i \leq 0$, $1 \leq i \leq N$. If the server node u has description i we

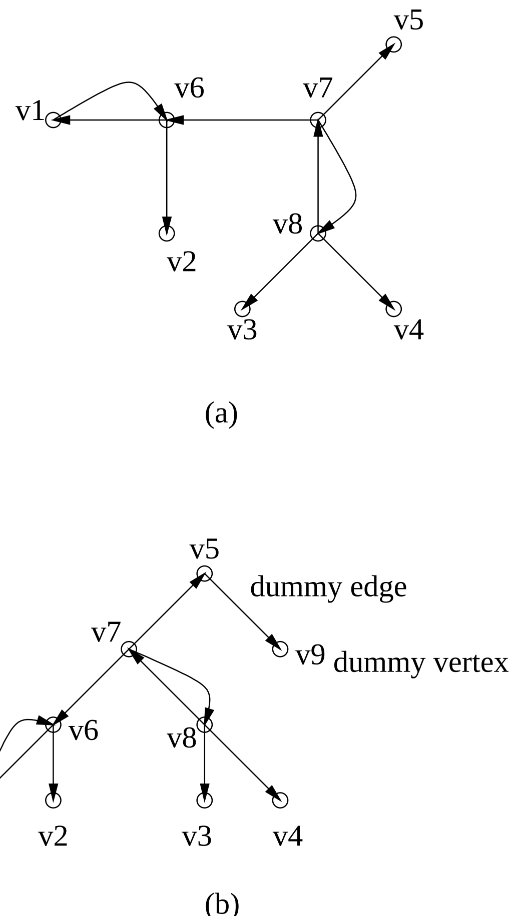

Fig. 4. (a) An example of a flow network that has a tree topology. (b) By adding a dummy vertex and a dummy edge, the tree topology can be regarded as rooted.

allow $f_i < 0$ so that u can send out the description; otherwise, $f_i = 0$. At the same time, $f_i < 0$ must satisfy the capacity constraint at the directed edge (u, u'). To summarize, $f_1, f_2, \ldots, f_N$ must satisfy the following conditions

1) $f_i \leq 0$ if $i \in \Psi_S(u)$,
2) $f_i = 0$ if $i \notin \Psi_S(u)$,
3) $\sum_{j:f_j<0} -f_j R_j \leq C(u, u')$.

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

$$DP(u, f_1, f_2, \ldots, f_N) = \begin{cases} 0, \text{ if } (f_1, f_2, \ldots, f_N) \in \mathcal{F}_1 \\ -\infty, \text{ if } (f_1, f_2, \ldots, f_N) \notin \mathcal{F}_1 \end{cases} \tag{9}$$

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

In this case, the node u can be a server, a sink, or both (e.g., in peer-to-peer networks). The flow volumes

$f_1, \ldots f_N$ in a feasible solution must satisfy the following constraints:

1) $f_i \leq 0$ if $i \in \Psi_S(u)$, $u \in S$,
2) $0 \leq f_i \leq 1$ if $u \notin S$ or if $u \in S$ but $i \notin \Psi_S(u)$.
3) $\sum_{j:f_j<0} -f_j R_j \leq C(u, u')$,
4) $\sum_{j:f_j>0} f_j R_j \leq C(u', u)$.

The first constraint is for when $u \in S$ has description i and does not need to receive it from its parent; the second constraint is to prevent u from receiving multiple copies of the same description, which simply wastes bandwidth without any coding gain; and the third and fourth conditions are to respect the edge capacity.

Let $\mathcal{F}_2$ be the set of vectors $(f_1, f_2, \ldots, f_N)$ that satisfy the above conditions. For each $f_i > 0$, u receives description i. Therefore,

$$DP(u, f_1, f_2, \ldots, f_N) = \begin{cases} F(\{i \,|\, f_i > 0 \text{ or } u \in S_i\}), \text{ if } (f_1, f_2, \ldots, f_N) \in \mathcal{F}_2 \\ -\infty, \text{ if } (f_1, f_2, \ldots, f_N) \notin \mathcal{F}_2 \end{cases} \tag{10}$$

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

Here, the node u can be a server, a router, or both. The capacity of the edges (u', u) and (u, u') require the flow volumes $f_1, \ldots f_N$ to satisfy the following constraints:

1) $f_i \leq 0$ if $i \in \Psi_S(u)$, $u \in S$,
2) $\sum_{j:f_j<0} -f_j R_j \leq C(u, u')$,
3) $\sum_{j:f_j>0} f_j R_j \leq C(u', u)$.

Let $\mathcal{F}_3$ be the set of vectors $(f_1, f_2, \ldots, f_N)$ that meet these conditions.

Let v and w be the two child nodes of u. Let f_i' and f_i'' be the number of i-colored paths passing from u to v and w, respectively. If u is not a server 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 server of color i, then f_i' and f_i'' can be flexible.

Therefore, for any $(f_1, \ldots, f_N) \notin \mathcal{F}_3$, $DP(u, f_1, \ldots, f_N) = -\infty$. And for any $(f_1, \ldots, f_N) \in \mathcal{F}_3$, we have

$$DP(u, f_1, \ldots, f_N) = \max(DP(v, f_1', f_2', \ldots, f_N') + DP(w, f_1'', f_2'', \ldots, f_N'')) \tag{11}$$

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

$$f_i' + f_i'' = f_i \quad \text{or} \quad i \in \Psi_S(u) \tag{12}$$

Case 4. u is an internal node and a sink.

This is the most general case where u is not only a sink, it may also be a server, a router, or all of the above. Like in Case 3, the flow volumes $f_1, \ldots, f_N$ have to be bounded by the capacities of edges (u', u) and (u, u'). Thus, any feasible solution requires $(f_1, \ldots, f_N) \in \mathcal{F}_3$.

The difference between Case 4 and Case 3 is that u can now reconstruct the set of MDC descriptions at its disposal, and hence contribute to the total fidelity. If u acts also as a relay, it cannot duplicate any received color i, thus $f_i = f_i' + f_i'' + 1$, with the exception that $u \in S$ and $i \in \Psi_S(u)$. Therefore, the recurrence of (11) is changed to

$$\begin{aligned} DP(u, f_1, \ldots, f_N) = \max(& DP(v, f_1', f_2', \ldots, f_N') + DP(w, f_1'', f_2'', \ldots, f_N'') \\ & + F(\{i \,|\, f_i = f_i' + f_i'' + 1 \text{ or } u \in S_i\})) \end{aligned} \tag{13}$$

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

$$f_i - 1 \leq f_i' + f_i'' \leq f_i \quad \text{or} \quad i \in \Psi_S(u). \tag{14}$$

Now we present a dynamic programming algorithm that computes the optimal RNF solution $DP(v_o, 0, \ldots, 0)$, with v_o being the root of tree network G. In post-order traversal of the tree nodes from the leaves to the root, the algorithm recursively solves the RNF problem for the subtrees rooted at the traversed nodes. At each step the current node u is classified into one of the above four cases. The corresponding recurrence relation is used to compute the optimal RNF solution $DP(u, f_1, f_2, \ldots, f_N)$ for the subtree G_u rooted at u from the optimal solutions $DP(v, f_1', f_2', \ldots, f_N')$ and $DP(w, f_1'', f_2'', \ldots, f_N'')$ for the left and right subtrees of G_u. The pseudocode of the algorithm is given below. In line 2 of the pseudocode, the range of the flow volume f_i is $|f_i| < |T|$ because each sink requires at most one copy of description i. Hear T is the set of sinks.

Algorithm Tree-RNF

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

Theorem 8: Algorithm Tree-RNF finds an optimal solution of RNF problem for networks of tree topology. Its time complexity is $O(|V| \times 4^N \times |T|^{2N})$.

Proof: The correctness of the algorithm follows from the above derivations for the all four possible cases. We only need to analyze the time complexity. The for loop at line 2 is repeated $O((2|T|)^N)$ times. Inside the for loop, the most time consuming step is line 14.

In (13), if $u \in S$ and $i \in \Psi_S(u)$, then there is no constraint on f_i' and f_i'' . Thus the maximizations over f_i' and f_i'', $max_{f_i'} DP(v, f_1', f_2', \ldots, f_N')$ and $max_{f_i''} DP(v, f_1'', f_2'', \ldots, f_N'')$, can be performed independently instead of in combinations of f_i' and f_i''. The most expensive case is when $u \notin S$. Therefore we only need to analyze the time complexity on this case. There are at most $|T|^N$ choices of $f_1', \ldots, f_N'$. Once $f_1, \ldots, f_N$ and $f_1', \ldots, f_N'$ are fixed, there are at most 2^N choices of $f_1'', \ldots, f_N''$ by the condition of (14). So the complexity of evaluating (13) is $O(|T|^N \times 2^N)$.

The loop in line 2 iterates $O((2|T|)^N)$ times, hence the cost of steps 2 through 14 is $O(|T|^{2N} 4^N)$ per tree node, resulting in the total complexity of $O(|V| 4^N |T|^{2N})$, which is polynomial if N is a constant. ∎

Clearly, the same algorithm can also compute optimal multiple-sink CRNF for tree network topology 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 CRNF.

Theorem 9: If $F(\mathcal{X}) = |\mathcal{X}|$, then the time complexity of Algorithm Tree-RNF can be reduced to $O(|V| \times |T|^{2N})$.

Proof: We only need to change the procedure at step 14. Because our goal is to maximize the total number

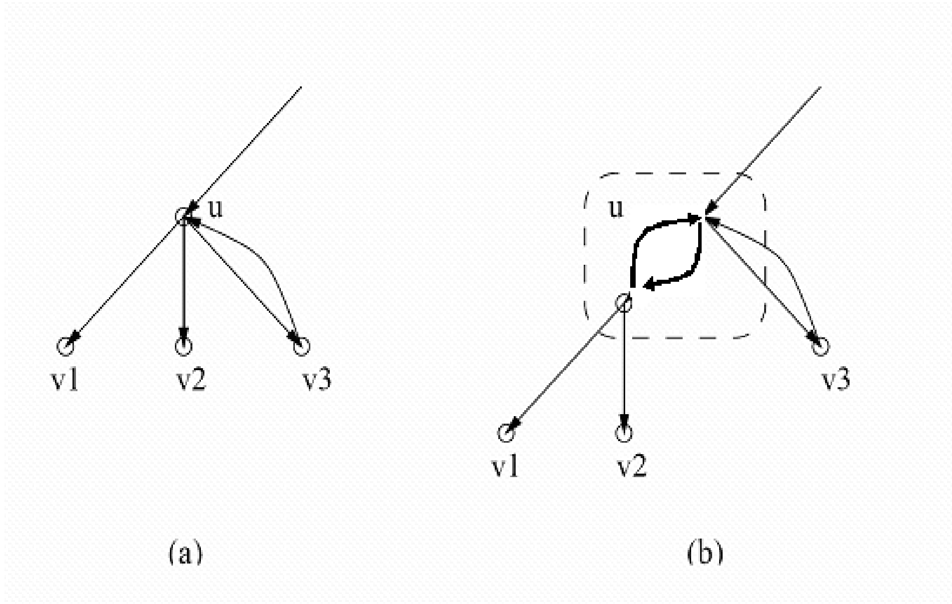

Fig. 5. A general tree can be reduced to a binary tree.

of distinct descriptions, it is suboptimal to have a path of color i passing through a sink u but no path of color i ending at u. So, we change the constraint in (14) to the following:

$$f_i - 1 = f_i' + f_i'' \text{ if } u \notin S \text{ or if } u \in S \text{ but } i \notin \Psi_S(u). \tag{15}$$

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

For trees with degree greater than 3, we can convert it to a binary tree. Each degree d node is split into $d-2$ nodes, the edges connecting these nodes have unlimited capacity. An example is given in Fig. 5. 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 trees as well.

VI. Multicast Rainbow Network Flow in Trees

As defined in (6), in multicast rainbow network problem (RNF*), the relay nodes are allowed to duplicate received colors and multicast them via selected links. RNF* uses edge capacities far more efficiently, but it also makes optimal routing of MDC descriptions in a general graph computationally very hard. However, if the network has a tree topology, the dynamic programming principle of Algorithm Tree-RNF clearly holds for RNF* as well. If the algorithm is modified according to the edge capacity constraint (6), it can compute optimal RNF* in polynomial time. In fact, it turns out that the tree topology makes optimal routing easier for RNF* than for RNF.

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 in an optimal solution of RNF*, there can be only one copy of the same description to pass through an edge.

The second modification is in Case 3. If u does not have description i, the algorithm considers two possibilities: (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, (12) is modified to

$$f_i = f'_i = f''_i = 0 \text{ or } f_i = 1 \text{ or } f'_i = -1 \text{ or } f''_i = -1 \text{ or } i \in \Psi_S(u), u \in S. \tag{16}$$

Likewise, the third modification is needed for Case 4. If u receives description i, then one of $f_i = 1$ or $f'_i = -1$ or $f''_i = -1$ is true. As a result, the computation in (13) is changed to

$$\begin{aligned} DP(u, f_1, \ldots, f_N) = \max(&DP(v, f'_1, f'_2, \ldots, f'_N) + DP(w, f''_1, f''_2, \ldots, f''_N) \\ &+F(\{i \,|\, f_i = 1 \text{ or } f'_i = -1 \text{ or } f''_i = -1 \text{ or } i \in \Psi_S(u), u \in S\}) \end{aligned} \tag{17}$$

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

$$f_i = f'_i = f''_i = 0 \text{ or } f_i = 1 \text{ or } f'_i = -1 \text{ or } f''_i = -1 \text{ or } i \in \Psi_S(u), u \in S. \tag{18}$$

These three modifications make Algorithm Tree-RNF to meet the edge capacity constraint (6) for RNF*. Then the algorithm can be used to compute an optimal solution of RNF* for tree topology.

The time complexity of the modified algorithm is dominated by the cost of the maximization task of (17). If nodes can multicast a description, then only one of $f_i = 1$, $f'_i = -1$, $f''_i = -1$ can hold in an optimal solution. With this property, when a node u is not a server of description i, the number of feasible choices of (f_i, f'_i, f''_i) restricted by (18) is only 13, as shown in Fig. 6.

As a result, a careful implementation of Algorithm Tree-RNF runs in $O(|V| \times 13^N)$ time, where N is the number of different MDC descriptions, which is a constant. Comparing this with Theorem 8, we see that the flow optimization actually has a significantly lower complexity for RNF* than for RNF, if the network has a tree topology. This is a particularly interesting observation in an algorithmic perspective. Here is a case that coding capability of nodes makes optimal routing an easier computation task than without, if duplicating and multicasting a data packet by a node is viewed as a special form of network coding.

VII. Optimal RNF for Full Source Spectrum

In formulating the RNF problem, we allow arbitrary color distributions at the server nodes. In general a server node does not supply all the descriptions of an MDC. This formulation models the situations in sensor networks and

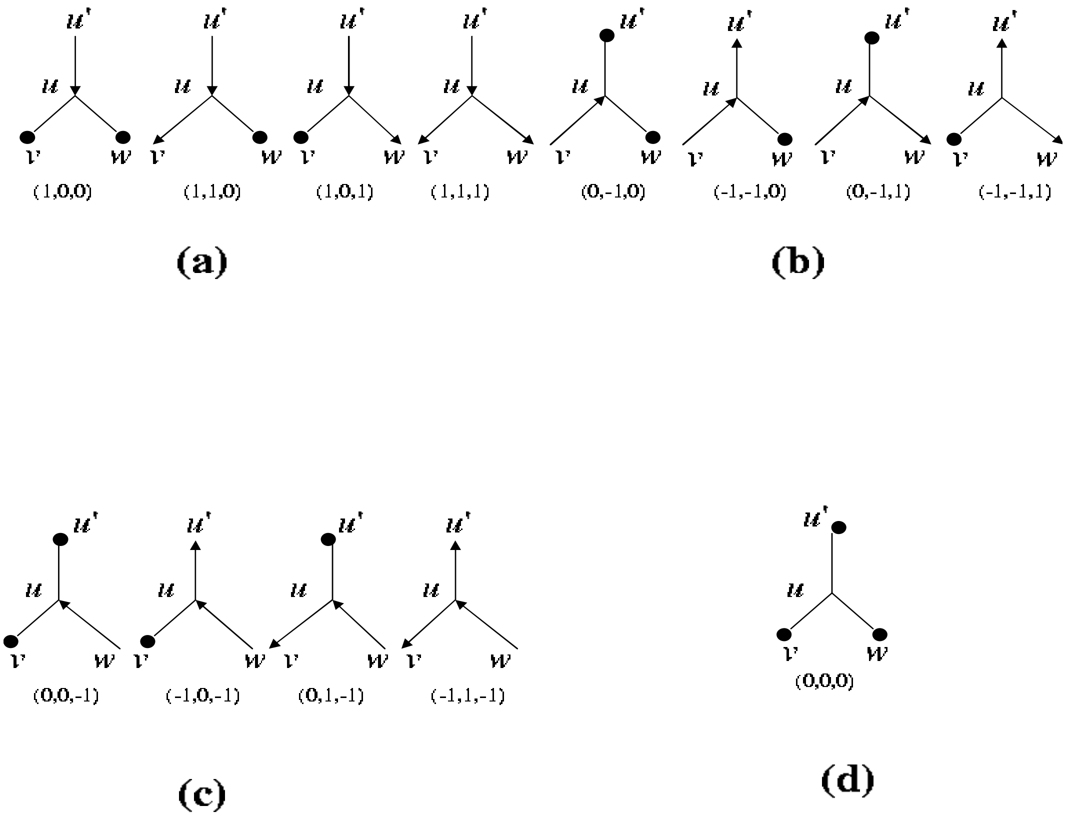

Fig. 6. The thirteen possible values of (f_i, f_i', f_i'') satisfying (18) and corresponding flow patterns.

distributed source coding where a server node (encoder) can only produce a subset of MDC descriptions. However, for computer network applications, in particular networked multimedia communications, it is straightforward and commonplace to code a signal in MDC and place all descriptions on the servers ahead of time. We call this scenario full source spectrum, i.e., $\Psi_S(s_i) = \{1, 2, \ldots, N\}$ for every server node $s_i \in S$. As we will see in this section, full source spectrum offers a way to trade the content redundancy and storage space for the tractability of many optimal RNF solutions.

In what follows we assume $w(t) = 1$. Consider the CRNF problem for full source spectrum. By a modification of the original network graph G, the CRNF problem can converted to and solved as one of monochrome maximum network flow. Specifically, we add a super server node s_o to network graph G, and direct an edge of infinite capacity from s_o to each server node $s_i \in S$. Also, we add a super sink node t_o to G, and direct an edge of capacity $C(t_i, t_o) = N$ from each sink node $t_i \in T$ to t_o. The resulting expanded graph is depicted in Fig. 7.

The next step is to compute a maximum monochrome flow from s_o to t_o using an existing maximum network flow algorithm (e.g., the one in [3]). By the construction of the expanded graph, any path in the computed maximum flow from s_o to t_o has to travel from a server node $s \in S$ to a sink node $t \in T$. Therefore, to each sink node

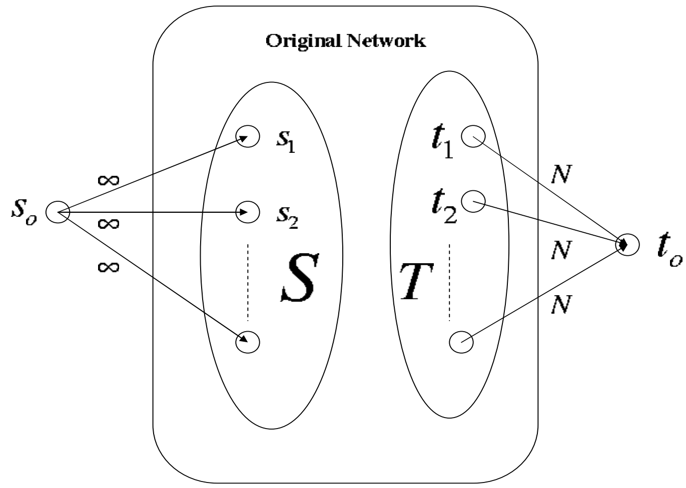

Fig. 7. The process of adding a super node s_o and a super sink t_o.

$t_i \in T$, $1 \leq i \leq |T|$, there will be a set $ß_i$ of flow paths from some server node(s). An optimal solution of the CRNF problem can be derived from these $|T|$ sets of paths. For every sink $t_i \in T$, we assign each path $p \in ß_i$ a distinct color k, $1 \leq k \leq |ß_i|$. Whatever server node $s \in S$ that emits this path of color k has the color since the spectrum $\Psi_S(s)$ is full.

The paths colored as such, $\sum_{1 \leq i \leq |T|} |ß_i|$ of them in total, constitute an optimal solution of the original CRNF problem, because $\sum_{1 \leq i \leq |T|} |ß_i|$ is the maximum flow volume allowed by the network from all servers to all sinks even without concerning distinct coloring of all incoming paths to any individual sink.

Note that the optimality of the above CRNF solution holds only if the relay nodes in the network cannot duplicate any received color. In practice, there are cases where duplicating an MDC description is not possible or desirable. This is particularly true in today's Internet where multicast backbone is not yet widespread. In this case the system cannot rely on multicast functionality provided by lower layer routers (see [2] for a detailed discussion). Also, a router may not have sufficient buffer space to hold an entire MDC description. Other reasons for not duplicating an MDC description in multicasting may be for data security or/and digital right protection. For instances, one does not want any router to know what clients are receiving an identical data stream, or the same content is differently encrypted or watermarked so that different copies are not interchangeable.

Finally, it should be noted that the solution for RNF provides a lower bound in performance for RNF*.

In the previous sections, when the source spectrum is not full, we witnessed the hardness of optimal RNF problem even for single sink and even when the fidelity function is monotone in the total rate. Remarkably, aided by the full source spectrum the optimization problem becomes far more amenable. Let us make a minor modification to the expanded graph in Fig. 7 by changing the edge capacity $C(t_i, t)$ from N to R_{max}, where $R_{max} = \sum_{1 \leq i \leq N} R_i$.

As in the case of CRNF, we can solve the equivalent problem of maximum monochrome flow from s_o to t_o, and determine the maximum possible flow volume K_i into each sink $t_i \in T$, $1 \leq i \leq |T|$, even allowing the delivery path of any MDC description to be split. Since every server node has full spectrum, we have the complete freedom with the coloring of the flow paths. Upon maximizing the total monochrome flow into all sinks, we now come to the following flow optimization problem in color:

$$\max_{\Psi \in \mathcal{K}^*} F(\Psi) \tag{19}$$
$$\text{subject to} \sum_{\kappa \in \Psi} R_\kappa \leq K_i,$$

independently for all $t_i \in T$, where $\mathcal{K}^*$ is the power set of all colors.

For full source spectrum, the reduced optimal RNF problem of (20) is polynomially solvable if the MDC in question is a scalable (layered) code. Indeed, upon having the maximum monochrome flow from s_o to t_o, for each sink $t \in T$, we let the monochrome flow volume L into sink t be the beginning consecutive portion of the scalable code, i.e., find the largest K such that $\sum_{1 \leq i \leq K} R_i \leq L$. This simple scheme is optimal since a scalable code (a special form of MDC) is only successively decodable, i.e., a sink node cannot decode description i unless it has all of descriptions $i-1, i-2, \ldots, 1$. Furthermore, if the scalable code is continuously scalable (the code stream can be truncated anywhere), such as the well-known SPHIT and JPEG 2000 image codes [12], [14], then all bits in the flow volume L into any sink $t \in T$ can be utilized by the above algorithm.

In contrast, if the source spectrum is not full, the dependency of the fidelity function $F(\Psi)$ on a particular order of received descriptions becomes a liability. It makes optimal RNF for scalable MDC computationally extremely hard.

If the fidelity function $F(\Psi)$ is additive, namely

$$F(\Psi) = \sum_{\kappa \in \Psi} F(\{\kappa\}), \tag{20}$$

then the optimization problem of (20) is a knapsack problem that can be solved in $O(N^2 K_i)$ time for each sink t_i of incoming flow volume of K_i. In this case, the optimal RNF problem can be solved in $O(N^2 K)$ time, where K is the volume of the maximum monochrome flow from s_o to t_o.

Recall from the previous section that optimal RNF problem is not computationally tractable even when the fidelity function $F(\Psi)$ is monotone in the total rate of all the descriptions in Ψ (the condition of (8)). However, the situation changes for full source spectrum. From (20) the optimal RNF problem for full source spectrum can be stated as

$$\max_{\Psi \in \mathcal{K}^*} \sum_{\kappa \in \Psi} R_\kappa \tag{21}$$
$$\text{subject to} \sum_{\kappa \in \Psi} R_\kappa \leq K_i,$$

independently for each $t_i \in T$. Again, this is a knapsack problem, in fact, a special case where the value of the item (description) to be packed is proportional to its size (rate). As in the instance of scalable code, the optimal RNF problem can be solved in $O(N^2 K)$ time. The problem is greatly simplified by the fact that a sink can fetch any description from any server node thanks to full source spectrum. In fact, full source spectrum allows an MDC description to be transmitted in different parts via different paths from one or more servers to a sink.

VIII. Conclusions and Future Work

We pose the rainbow network flow problem of routing MDC descriptions to maximize a collective fidelity metric over all sinks, constrained by the edge capacities. This problem is proven to be Max-SNP hard for general network topology and an arbitrary distribution of MDC descriptions, and hence it is unlikely to have a polynomial-time approximation scheme that approaches the optimal solution arbitrarily close. But this negative result does not preclude polynomial-time solutions of the optimal RNF problem in many practically important cases. Algorithms of computing optimal RNF for single sink, full source spectrum, and tree network topology are presented, and their complexities are analyzed.

Interesting and challenging problems for future research on RNF are many. In a constructive perspective, the most important topic is perhaps how to jointly optimize the design and routing of MDC for a network. Information theoretically, an intriguing problem is to find the achievable region for the paradigm of lossy joint source-network coding for multiple servers and multiple sinks.

Given the NP-hardness of optimal RNF for general networks, good heuristic practical solutions are in order. This paper showed that optimal RNF problem can be efficiently solved with respect to a single sink in general graphs. Then can we find a good RNF solution for multiple sinks by a clever use of this algorithm for single sink?

We also showed how full source spectrum can make some variants of RNF problem computationally more amenable. Then a related problem is how to strategically place MDC descriptions at the servers if there are storage capacity limits at the servers. Even more challengingly, how should a given number of servers (with or without storage capacities) be placed among a set of possible sites in the network in a rate-distortion optimal way?

Appendix A. Proof of Theorem 1

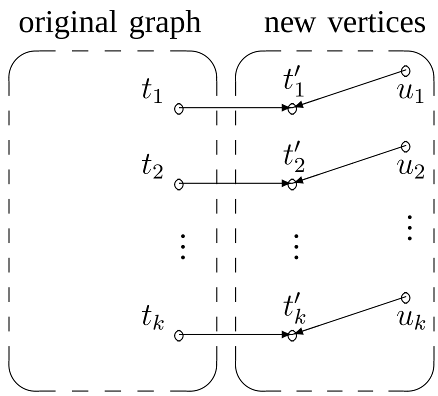

Fig. 8. Two new vertices, t_i' and u_i, are added for each t_i. u_i has all but color i. Edge (u_i, t_i) has capacity $k-1$. All the other edges have capacity 1.

We will show that the CRNF problem can be reduced to from the known NP-hard problem, maximum edge disjoint paths [9], [6], which is stated below.

Maximum edge disjoint paths problem. Given a graph $G = \langle V, E \rangle$ and k pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$, find k edge disjoint paths, each connecting a different vertex pair s_i, t_i.

Proof: Let $G = \langle V, E \rangle$ be an instance of maximum edge disjoint paths with k pairs of nodes (s_i, t_i). We construct an instance for CRNF as follows.

There are k different colors. For each t_i, construct two new vertices t_i' and u_i. Add a direct edge (t_i, t_i') with capacity 1 and a direct edge (u_i, t_i') with capacity $k-1$ (See Fig. 8). The sink set $T = \{t_1', \ldots, t_k'\}$. Assign all the edges in E with capacity 1. For each color i, the corresponding server vertex set $S_i = \{s_i\} \cup \{u_j \mid j \neq i\}$. We get an instance of CRNF problem.

If the original maximum edge disjoint path instance has a solution, suppose p_i connects from s_i to t_i. Then we can construct a solution of CRNF with total income k^2 as follows. For each color i, $P_i = \{(u_j, t_j') \mid j \neq i\} \cup \{p_i \cdot (t_i, t_i')\}$.

Conversely, suppose the CRNF problem has a solution with total income k^2. Then there are k paths with k different colors ending at each t_i'. Because the only two incoming edges for t_i' are (t_i, t_i') with capacity 1 and (u_i, t_i') with capacity $k-1$, for each i there must be a path of color i that starts at s_i and passes t_i. Such paths for different i must be edge-disjoint because all the edges in E have capacity only 1. Hence we get a solution for the maximum edge disjoint path problem.

Therefore, maximum edge disjoint path is reduced to CRNF. Because the former is NP-hard, so is the latter.

Finally, we claim that CRNF problem remains NP-hard even if the network graph is a directed acyclic planar graph. This follows from the fact that the maximum directed edge disjoint paths problem is NP-complete even if

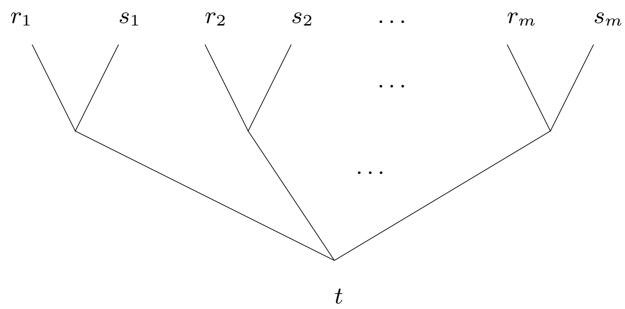

Fig. 9. The topology of the constructed rainbow network flow in the proof of Theorem 3.

the underlying graph is acyclic planar [15].

■

Appendix B. Proof of Theorem 3

Our proof is based on the fact that the Max-2-SAT problem is Max-SNP-hard [11], which is defined below.

Max-2-SAT problem 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.

Proof: We construct an L-reduction [11] from Max-2-SAT to the RNF 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 the RNF problem has $2n$ descriptions, each corresponds to either x_i or $\bar{x}_i$. It also has $2n$ server nodes. The server nodes are grouped into n pairs, r_i and s_i, $i = 1, 2, \ldots, n$. Each r_i has the description corresponding to x_i, and each s_i has the description corresponding to $\bar{x}_i$.

Then all the nodes are connected as in Fig. 9. There is only one sink node t. Each description has a fixed unit rate. And the capacity of each edge is also one in that unit. This graph ensures that only one of the two descriptions x_i and $\bar{x}_i$ can reach the sink t.

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

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

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

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

Thus, we have constructed an instance, $\mathcal{J}$, of the RNF 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, the RNF problem is also Max-SNP-hard. ■

Appendix C. Proof of Theorem 5

Proof: In the proof in Appendix A, each t'_i needs and only needs to receive a different color i from t_i. Therefore, there will be only one copy of each color transporting in the original graph. (See Fig. 8) The extra ability of duplicating a color does not change the solution. Therefore, the reduction works for CRNF* as well.

Similarly, it is obvious that the extra ability of duplicating a color does not change the solution in the proof in Appendix B. Therefore, the reduction in Appendix B works for RNF* as well.

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] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE J. on Selected Areas in Comm.*, vol. 20, no. 8, Oct 2002.
[3] A. Goldberg and R. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM*, vol. 35, no. 4, pp. 921–940, 1988.
[4] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, 2001.
[5] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proc. Information Processing in Sensor Networks 2004*.
[6] R. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, pp. 45–68, 1975.
[7] V. King, S. Rao, and R. Tarjan, "Faster deterministic maximum flow algorithm," in *Symposium on Discrete Algorithms*, 1992, pp. 157–164.
[8] J. M. Kleinberg, "Single-source unsplittable flow," in *37th IEEE Symp. on Foundations of Computer Science*, 1996, pp. 68–77.
[9] R. Miller and J. Thatcher, Eds., *Complexity of Computations*. New York: Plenum Press, 1972.
[10] L. H. Ozarow, "On a source coding problem with two channels and three receivers," *Bell System Technical Journal, 59(10)*, 1980.
[11] C. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *Journal Computer and System Sciences*, vol. 43, pp. 425–440, 1991.
[12] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 6, pp. 243–250, 1996.
[13] N. Sarshar and X. Wu, "A practical approach to joint network source coding," *Proc. of Data Compression Conference*, 2006.
[14] A. Taubman and M. Marcellin, "Jpeg2000: Image compression fundamentals, practice and standard," *Kluwer Academic Publishers*, 2001.
[15] J. Vygen, "Np-completeness of some edge-disjoint paths problems," *Discrete Applied Mathematics*, vol. 61(1), pp. 83–90, 1995.