Abstract — We propose a subtree decomposition method for network coding. This approach enables derivation of tight bounds on the network code alphabet size, makes connections with convolutional codes transparent, allows specification of coding operations at network nodes without the knowledge of the overall network topology, and facilitates design of codes which can easily accommodate future changes in the network, such as addition of receivers and loss of links.

I. SUBTREE DECOMPOSITION METHOD

We represent a multicast communication network as an acyclic directed graph $G = (V,E)$ with unit capacity edges, $h$ unit-rate, co-located, information sources $\{S_1, \ldots, S_h\}$ simultaneously sending information to $N$ receivers $\{R_1, \ldots, R_N\}$. The number of edges of the min-cut between the source and each receiver node is at least $h$. Under these conditions, there exists a linear network multicast code [1].

Subtree decomposition refers to partitioning the line graph of the network into a disjoint union of connected subsets $\{T_i\}$ through which the same information flows. Each $T_i$ is a tree, hence the name of the method. For the network code design problem, we only need to know how the subtrees are connected and which receivers observe each $T_i$, whereas the structure of the network inside a subtree does not play any role. Thus we can contract each subtree to a node and retain only the edges that connect the subtrees, to get the subtree graph. A detailed description of this procedure is provided in [2, 3].

Figure 1 depicts an example of a network with two sources multicasting to the same set of three receivers, and its subtree graph. The min-cut condition in the network $G$ is equivalent to the following condition in the subtree graph: For each receiver $R_j$, the $h$ paths from the $h$ source nodes to the $h$ receiver $R_j$ nodes are both edge and vertex disjoint. We will call this property the multicast property in the subtree graph.

Definition 1 A subtree graph is called minimal if removing any edge would violate the multicast property.

II. APPLICATIONS

The subtree decomposition can be thought of as a method to extract from the network only the information that is necessary for the network code design. The method enables classifications of multicast configurations into equivalence classes, where two topologies are equivalent if they lead to the same subtree graph. For example, it is easy to show the following:

Theorem 1 For a network with two sources and two receivers, there exist exactly two different minimal subtree graphs.

A. Alphabet Size Bounds

We can use the properties of minimal subtree graphs to derive alphabet size bounds. For example, we have the following:

Theorem 2 For any minimal configuration with $h = 2$ sources and $N$ receivers, the alphabet $F_q$ of size

$$q \leq \left\lfloor \frac{2N - 7/4}{1/2} \right\rfloor$$

is sufficient. There exist configurations for which this alphabet is necessary.

B. Decentralized Algorithms

We can use the subtree decomposition to design decentralized algorithms that are valid for a number of network topologies. Network coding amounts to associating a vector of linear coefficients, called coding vector, with each subtree [3]. In designing a network code, we need to ensure that the coding vectors assigned to the subtrees having receivers in common are linearly independent. Selecting coding vectors that belong to a geometric object known as arc, for the configurations where this is possible, automatically ensures this condition is satisfied [9].

C. Scalability

An important practical consideration is the scalability of a network code to the addition of new users. One of the main advantages of decentralized codes is that they do not have to be changed with the addition of new receivers as long as its subtree decomposition of the network remains the same. In general, addition of new users does not change the network code for the existing users as long as the new subtree graph contains the original subtree graph.

ACKNOWLEDGMENTS

We acknowledge the support of DIMACS, NSF under award No. CCR-0325673, and FNS under award No. 200021-103836/1. We also thank D. Ron and E. Esichen for useful feedback.

REFERENCES

