An Iterative Framework for Optimizing Multicast Throughput in Wireless Networks

Lihua Wan and Jie Luo
Electrical & Computer Engineering Department
Colorado State University
Fort Collins, CO 80523
Email: {carlawan, rockey}@engr.colostate.edu

Anthony Ephremides
Electrical & Computer Engineering Department
University of Maryland
College Park, MD 20742
Email: etony@umd.edu

***Abstract*— This paper shows that, under certain conditions, a wireless network can be modeled by a directed configuration graph with possible hyperarc links if the transmission schedule is given. Assume single multicast session. The maximum achievable multicast throughput equals the max-flow min-cut bound of the configuration graph. An optimization framework is proposed to maximize the multicast throughput via iterative updates of the transmission schedule. It is demonstrated that the optimal multicast throughput can be obtained without exploring either a large number of hyperarc links or a large number of cuts, although efficient suboptimal algorithm is needed to avoid searching link combinations and to reduce the complexity further to polynomial in the number of nodes. It is also shown that, when the configuration graph has hyperarc links, the minimum cut can no longer be obtained using the well-known flow augmenting path algorithm. An alternative algorithm is proposed.**[1]

I. Introduction

The topology of a wireline network can be modeled by a graph, in which, each vertex represents a network node and each edge represents a point-to-point (cable) link between two nodes. Extending this graphic model to a wireless network faces two key challenges. First, since signal transmitted over the wireless medium can often reach more than one receivers, it is possible that a wireless node can communicate common information to multiple receivers simultaneously [1] using the same power and bandwidth of a point-to-point transmission. To model such (direct) multicast transmission, it is necessary that a graph representation should contain point-to-multipoint edges (termed hyperarc links [2]). Meanwhile, a wireless link can support a positive information rate so long as the channel gain is not strictly zero. If all links with nonzero channel gains must be included in a graph representation, the total number of links is exponential in the number of nodes[2]. Consequently, optimizing all link rates simultaneously can be overly complex even for a moderate-sized network. Second, communication over a wireless link can be interfered by signals transmitted from the neighboring nodes. Communication rates of the links are therefore coupled. If we list the rates of all links as a vector of dimension $|E|$, then the closure of all achievable rate vectors is characterized by a *capacity region* in the $|E|$-dimensional space. Unfortunately, obtaining the capacity region, or testifying whether a given rate vector is in the capacity region, can be extremely difficult even for a small network with three or four nodes [3].

Advances of network coding showed that, if a source node transmits common information to multiple destinations in a wireline network, the maximum achievable multicast throughput equals the maximum flow of the minimum cut that separates the source from at least one destination in the topology graph [4]. This result has stimulated a series of consequential researches on optimizing multicast throughput in wireless and mesh networks [2][5][6][7].

In this paper, we propose an iterative framework to maximize the throughput of a single multicast session in a wireless network. We show that, given the transmission schedule (defined in Section III) and under certain conditions, a wireless network can be represented by a configuration graph with possible hyperarc links. Given the configuration graph, with the help of network coding, the multicast throughput equals the maximum flow of the minimum cut that separates the source node from at least one destination node. The key idea of the framework is therefore to iteratively update the transmission schedule to improve the max-flow min-cut value of the corresponding configuration graph. We show that the proposed framework can address both key challenges mentioned at the beginning of this section. Efficient network configurations can also be obtained with a complexity polynomial in the number of nodes.

II. Problem Formulations

Let V be the node set of a wireless network. We consider a single multicast session where the source node $s \in V$ delivers common information *reliably* (in information

[1]This work was supported by National Science Foundation grants CCF 07-28826 and CCF 07-28966.

[2]For example, consider a network with $|V|$ nodes located in an open area. Since each node can transmit common information to any subset of nodes, the total number of links (including hyperarc links) equals $|V|2^{|V|-1}$.

theoretic sense), possibly through multi-hop paths, to *all* nodes in a destination set $T \subset V$. The information rate of such transmission is termed the multicast throughput, and is denoted by R_{sT}.

A *communication link*, e_{iJ}, is defined as the association of one tail (transmitter) node i and a set of head (receiver) nodes J. If we can list all nodes in J, for example $J = \{a, b\}$, we also denote the link e_{iJ} as $e_{i\overline{ab}}$. We say e_{iJ} achieves an information rate of r_{iJ} if i communicates common information reliably and *directly* to all nodes in J at rate r_{iJ}.

We assume the peak transmission power of node i must be kept below P_i. Let $\boldsymbol{P}$ be a $|V|$-dimensional column vector whose elements are the peak power bounds of the nodes. Let $\boldsymbol{r}$ be a column vector whose elements are the information rates of *all* feasible communication links. Assume channel gains between network nodes are time-invariant. The union of all achievable rate vectors $\boldsymbol{r}$, denoted by $\mathcal{C}_{\boldsymbol{r}}(\boldsymbol{P})$, is defined as the *link capacity region* of the wireless network, which is a function of $\boldsymbol{P}$. We assume the multicast throughput is a function of the link rate vector $\boldsymbol{r}$, and formulate the multicast throughput maximization problem as

$$\text{maximize } R_{sT}(\boldsymbol{r}), \qquad \text{s.t. } \boldsymbol{r} \in \mathcal{C}_{\boldsymbol{r}}(\boldsymbol{P}). \tag{1}$$

Note that by formulating the optimization problem (1), we have made two key assumptions. First, writing $R_{sT}(\boldsymbol{r})$ as a function of $\boldsymbol{r}$ assumes information should be transmitted reliably over each link. This assumption excludes the possible operation of amplify-and-forward at the relay nodes. Let us consider the four-node network illustrated in Figure 1a. We assume the connections between s and a, b

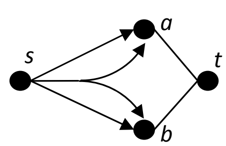

Fig. 1a. a, b, t are connected via wireline links. Joint decoding at t achieves a higher throughput than decoding information independently at a and b.

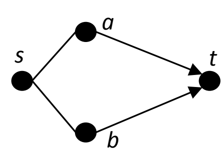

Fig. 1b. s, a, b are connected via wireline links. Jointly encoding information at a, b achieves a higher throughput than encoding information independently.

are wireless, while a, b and t are connected via noiseless wireline links. Assume s, a, b have single antenna each. The channel gains between s, a and s, b are denoted by h_{sa} and h_{sb}, respectively. Assume $h_{sb} > h_{sa}$. Let the ambient noise be white Gaussian with zero mean and variance N_0. If we regard a and b as two receiving antennas of t and jointly decode the message at t, the achievable information rate from s to t is given by

$$R_{st} = \frac{1}{2}\log\left(1 + (h_{sa}^2 + h_{sb}^2)\frac{P_s}{N_0}\right). \tag{2}$$

If we assume information must be reliably decoded at a and b independently, and then be forwarded to t, the maximum achievable rate from s to t is given by

$$R_{st} = \frac{1}{2}\log\left(1 + h_{sb}^2\frac{P_s}{N_0}\right), \tag{3}$$

which is less than the rate of (2).

Second, by presenting each communication link with a single tail node, we also exclude possible joint encoding of common information at different nodes. Let us consider the four-node network illustrated in Figure 1b. We assume s, a and b are connected via noiseless wireline links, while the connections between a, t and b, t are wireless ones. Let the channel gains between a, t and b, t be h_{at} and h_{bt}, respectively. Assume the ambient noise is white Gaussian with zero mean and variance N_0. If we regard a and b as two transmitting antennas of s, and jointly encode and transmit the message at these two nodes, then the achievable information rate from s to t is given by

$$R_{st} = \frac{1}{2}\log\left(1 + \frac{(|h_{at}|\sqrt{P_a} + |h_{bt}|\sqrt{P_b})^2}{N_0}\right). \tag{4}$$

If we assume the information must be encoded independently at a and b, the achievable information rate becomes

$$R_{st} = \frac{1}{2}\log\left(1 + \frac{(h_{at}^2 P_a + h_{bt}^2 P_b)}{N_0}\right), \tag{5}$$

which is less than the rate of (4).

In addition to the above two assumptions, in this paper, we also assume reliable communication over a link is achieved without channel feedback exploitation. Although making these assumptions may cause throughput loss, it enables us to represent a wireless network using a configuration graph given the transmission schedule. This consequently leads to an efficient cross-layer optimization framework, as explained in the next two sections.

III. Problem Decomposition

A *communication realization*, $C_k = \left\{\left(e_{iJ}, r_{iJ}^{(k)}\right)\right\}$, is defined as a set of link and rate pairs, where $r_{iJ}^{(k)}$ is the information rate over link e_{iJ}. We say $e_{iJ} \in C_k$ if $r_{iJ}^{(k)} > 0$, and let $r_{iJ}^{(k)} = 0$ if $e_{iJ} \notin C_k$. Define the corresponding link rate vector as $\boldsymbol{r}^{(k)}$. We say the communication realization C_k is *feasible* if $\boldsymbol{r}^{(k)}$ can be achieved without invoking the time sharing operation.

A *transmission schedule*, $S = \{(C_k, p_k)\}$, is defined as a set of feasible communication realization and time proportion pairs, where $\sum p_k = 1$, and $0 \leq p_k \leq 1$ is the time proportion when communication realization C_k is active. We say $C_k \in S$ if $p_k > 0$.

Given transmission schedule S, we can construct a directed *configuration graph* $G(S) = \{V, E\}$, where V and E are the node set and the edge set, respectively. $e_{iJ} \in E$ if we can find a $C_k \in S$ such that $e_{iJ} \in C_k$. We associate with each link in the configuration graph a *configuration rate* $g_{iJ}(S) = \sum_{C_k \in S} r_{iJ}^{(k)} p_k$. Note that, whenever we talk about a configuration graph, we always assume the transmission schedule is specified.

In the configuration graph, a *cut* γ_m is defined as a partition that divides the node set V into two disjoint subsets $V_l^{(m)}$ and $V_r^{(m)}$. We say γ_m is an $s-t_k$ cut if $s \in V_l^{(m)}$ and $t_k \in V_r^{(m)}$. We say γ_m is an $s-T$ cut if it is an $s-t_k$ cut for at least one $t_k \in T$. A link e_{iJ} crosses cut γ_m if the tail node i satisfies $i \in V_l^{(m)}$, and at least one head node $j \in J$ satisfies $j \in V_r^{(m)}$. The *cut value* of γ_m, also denoted by γ_m, is the sum configuration rates of all links crossing γ_m. Let $\boldsymbol{\gamma}(S)$ be a column vector whose elements are the values of all the $s{-}T$ cuts. Denote the m^{th} element of $\boldsymbol{\gamma}(S)$ by $[\boldsymbol{\gamma}(S)]_m$. As shown in [4], given S, via discarding and network coding information, the maximum achievable multicast throughput equals

$$R_{sT} = \min_m [\boldsymbol{\gamma}(S)]_m. \tag{6}$$

Example 1: Consider a three-node wireless network where s wants to deliver common information to two destination nodes t_1 and t_2. Assume we have three feasible communication realizations: $C_1 = \{(e_{st_1}, 4)\}$, $C_2 = \{(e_{st_2}, 4)\}$, $C_3 = \{(e_{s\overline{t_1 t_2}}, 3)\}$[3]. Given transmission schedule $S = \{(C_1, \frac{1}{3}), (C_2, \frac{1}{3}), (C_3, \frac{1}{3})\}$, we can form a configuration graph illustrated in Figure 2, in which the configuration rates are $g_{st_1} = \frac{4}{3}$, $g_{st_2} = \frac{4}{3}$, $g_{s\overline{t_1 t_2}} = 1$. Let $T = \{t_1, t_2\}$, the three $s-T$ cuts are illustrated by the dashed lines in Figure 2. The cut values are $\gamma_1 = \frac{7}{3}$, $\gamma_2 = \frac{7}{3}$, $\gamma_3 = \frac{11}{3}$, respectively. Given S, the maximum achievable multicast throughput equals $R_{\overline{st_1 t_2}} = \min(\gamma_1, \gamma_2, \gamma_3) = \frac{7}{3}$.

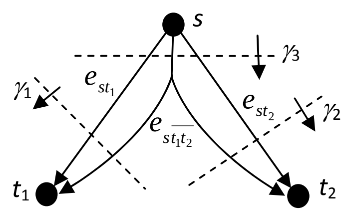

Fig. 2. A three-node network with three $s-T$ cuts.

Example 2: In this example, the network has four nodes. The source node s wants to transmit information to the destination node t. Assume we have the following three feasible communication realizations: $C_1 = \{(e_{s\overline{ab}}, 3)\}$, $C_2 = \{(e_{sb}, 3), (e_{at}, 3)\}$, $C_3 = \{(e_{bt}, 3)\}$. Given the transmission schedule $S = \{(C_1, \frac{1}{3}), (C_2, \frac{1}{3}), (C_3, \frac{1}{3})\}$, we can form a configuration graph illustrated in Figure 3, in which the configuration rates are $g_{s\overline{ab}} = 1$, $g_{sb} = 1$, $g_{at} = 1$, $g_{bt} = 1$. It is easy to verify that the maximum achievable throughput from s to t is $R_{sT} = 2$.

Let the union of $\boldsymbol{\gamma}(S)$ (taken over all S) be defined as the $s{-}T$ *cut capacity region* $\mathcal{C}_{\boldsymbol{\gamma}}(\boldsymbol{P})$. The optimization problem (1) can be rewritten as

$$\max_S \min_m [\boldsymbol{\gamma}(S)]_m, \qquad \text{s.t. } \boldsymbol{\gamma} \in \mathcal{C}_{\boldsymbol{\gamma}}(\boldsymbol{P}). \tag{7}$$

[3]These information rates can appear in a practical system if s is equipped with multiple antennas.

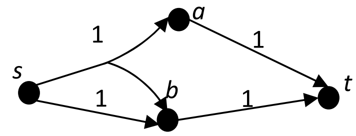

Fig. 3. A four-node network with link capacities.

IV. The Cross-layer Optimization Framework

Let $\boldsymbol{\lambda}$ be a column vector of the same dimension of $\boldsymbol{\gamma}$ with non-negative real-valued elements. Optimization problem (7) can be equivalently written as

$$\max_{\boldsymbol{\gamma} \in \mathcal{C}_{\boldsymbol{\gamma}}} \min_{\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq \mathbf{0}, \sum_m [\boldsymbol{\lambda}]_m = 1} \boldsymbol{\lambda}^T \boldsymbol{\gamma}. \tag{8}$$

Since (7) is a convex optimization problem, the throughput achieved by equilibriums of (8) must be unique. This optimal throughput can be obtained numerically via iteratively carrying out the following two steps.

The Basic Iterative Framework

Step 1: Update $\boldsymbol{\lambda}$ by $\boldsymbol{\lambda} = \boldsymbol{\lambda} - \delta_1 \boldsymbol{\gamma}$, where $\delta_1 > 0$ is the step size. Project $\boldsymbol{\lambda}$ to the constraint set to satisfy $\boldsymbol{\lambda} \geq \mathbf{0}$, $\sum_m [\boldsymbol{\lambda}]_m = 1$.

Step 2: Update $\boldsymbol{\gamma}$ by $\boldsymbol{\gamma} = \boldsymbol{\gamma} + \delta_2 \boldsymbol{\lambda}$, where $\delta_2 > 0$ is the step size. Project $\boldsymbol{\gamma}$ to the constraint set to satisfy $\boldsymbol{\gamma} \in \mathcal{C}_{\boldsymbol{\gamma}}$.

Unfortunately, one can rarely implement this basic algorithm in a practical system due to the two key challenges mentioned in Section I. Particularly, the closed-form expression of $\mathcal{C}_{\boldsymbol{\gamma}}$ is often not available; optimizing all $s-T$ cuts together can also be overly complex since the number of $s-T$ cuts is exponential in the number of nodes. In the following two sections, we show these difficulties can be addressed by revising the two steps correspondingly.

A. Revision on Step 1

To avoid optimizing all $s-T$ cuts together, it is necessary to upper bound the number of nonzeros elements in $\boldsymbol{\lambda}$. In this section, we show this can be done without sacrificing the optimality of the solution.

Let $|I_{\min}(\boldsymbol{\gamma})|$ denote the number of cuts that achieve the minimum value in $\boldsymbol{\gamma}$. Let Γ^* be the set of $s-T$ cut vectors that achieve the optimal throughput of (7). Define κ^* as the minimum number of minimum cuts among all the optimal $s-T$ cuts, i.e.,

$$\kappa^* = \min_{\boldsymbol{\gamma} \in \Gamma^*} |I_{\min}(\boldsymbol{\gamma})|. \tag{9}$$

Although we have $\kappa^* = 1$ for most of the wireline networks, the following proposition shows that $\kappa^* > 1$ often holds for multi-hop wireless networks.

Proposition 1: Let κ^* be defined by (9). If $\kappa^* = 1$, then R_{sT} is maximized by a transmission schedule S that contains only a single communication realization C ($S = \{(C, 1)\}$); the communication realization C only contains a single link e_{sT}, i.e., the source node s *directly* multicast common information to all destination nodes in T. ■

Let $\kappa > 0$ be an integer. For a given $\boldsymbol{\gamma}$, let $I_{\min}^{\kappa}(\boldsymbol{\gamma})$ be the set of indices corresponding to the first κ small-valued cuts in $\boldsymbol{\gamma}$. In other words, for all $m \notin I_{\min}^{\kappa}(\boldsymbol{\gamma})$ and $n \in I_{\min}^{\kappa}(\boldsymbol{\gamma})$, we have $[\boldsymbol{\gamma}]_m \geq [\boldsymbol{\gamma}]_n$. In the case when multiple cuts achieve the same cut value, we assume the choice of $I_{\min}^{\kappa}(\boldsymbol{\gamma})$ is deterministic with respect to $\boldsymbol{\gamma}$.

The following proposition shows that (8) can be written in another equivalent form.

Proposition 2: For all $\kappa \geq \kappa^*$, all equilibriums of the following optimization problem achieve the optimal throughput of (8),

$$\max_{\boldsymbol{\gamma} \in \mathcal{C}_{\boldsymbol{\gamma}}} \min_{\substack{\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq \mathbf{0}, \sum_m [\boldsymbol{\lambda}]_m = 1 \\ [\boldsymbol{\lambda}]_m = 0, \forall m \notin I_{\min}^{\kappa}(\boldsymbol{\gamma})}} \boldsymbol{\lambda}^T \boldsymbol{\gamma}. \tag{10}$$

■

Based on Proposition 2, given $\kappa \geq \kappa^*$, we can revise Step 1 of the iterative algorithm as follows.

Step 1 (revised): For all $m \in I_{\min}^{\kappa}(\boldsymbol{\gamma})$, update $[\boldsymbol{\lambda}]_m$ by $[\boldsymbol{\lambda}]_m = [\boldsymbol{\lambda}]_m - \delta_1 [\boldsymbol{\gamma}]_m$, where $\delta_1 > 0$ is the step size. For all $m \notin I_{\min}^{\kappa}(\boldsymbol{\gamma})$, set $[\boldsymbol{\lambda}]_m = 0$. Project $\boldsymbol{\lambda}$ to the constraint set to satisfy $\boldsymbol{\lambda} \geq \mathbf{0}$, $\sum_m [\boldsymbol{\lambda}]_m = 1$.

B. Revision on Step 2

Since the closed-form expression of $\mathcal{C}_{\boldsymbol{\gamma}}$ is often not available, it is necessary to keep track on the transmission schedule corresponding to the cut vector $\boldsymbol{\gamma}$ to ensure that $\boldsymbol{\gamma}$ is in the cut capacity region. Consequently, the task of Step 2 in the algorithm is to update the transmission schedule S such that $\boldsymbol{\lambda}^T \boldsymbol{\gamma}(S)$ can be improved. Since such update should be incremental, we can further assume the update is driven by a single communication realization in the sense that one should either add a new feasible communication realization into S, or increase the time proportion of an existing communication realization in S (and then scale the time proportions of other communication realizations in S accordingly).

Before implementing such revision to Step 2, we have to answer two key questions. First, whether improving $\boldsymbol{\lambda}^T \boldsymbol{\gamma}(S)$ is always possible by updating S with a single communication realization. Second, since the number of feasible links (including hyperarc links) can be exponential in the number of nodes, whether it is possible to explore only a polynomial number of links to find the optimal communication realization. Unfortunately, we can only get a positive answer to the two questions under the following additional assumption.

Assumption 1: For any communication realization, we assume information transmitted over a link is decoded without exploiting codebook information of other links[4].

[4]Take a multiaccess scheme for example. Assumption 1 prevents the use of joint multiuser decoding such as the maximum likelihood and the decision feedback multiuser detection algorithms. However, it does *not* exclude interference avoidance methods such as the decorrelation detection and the minimum mean square error (MMSE) detection, since these detectors only exploit the channel gain information of other links, but not their codebooks.

In the following proposition, we show that, under Assumption 1 (presented below), not only the transmission schedule update is always possible, we also do not need to activate any hyperarc link with more than κ^* receivers, where κ^* is defined in (9).

Proposition 3: Let Assumption 1 be enforced. Let κ^* (and $|I_{\min}(\boldsymbol{\gamma})|$) be defined by (9) under Assumption 1. For any feasible communication realization C, define $\boldsymbol{\gamma}(C)$ as the $s{-}T$ cut set vector corresponding to (the configuration graph derived from) transmission schedule $S = \{(C, 1)\}$.

Given $\boldsymbol{\lambda}$, and a cut vector $\boldsymbol{\gamma}$ corresponding to transmission schedule S. Assume $\boldsymbol{\lambda}^T \boldsymbol{\gamma}(S)$ is strictly less than the optimal throughput of (7). Then we can always find a communication realization C, which does not contain any hyperarc link with more than κ^* receivers, such that the following inequality is satisfied,

$$\boldsymbol{\lambda}^T \boldsymbol{\gamma}(C) > \boldsymbol{\lambda}^T \boldsymbol{\gamma}(S). \tag{11}$$

■

Based on Proposition 3, given $\kappa \geq \kappa^*$, we can now revise Step 2 of the iterative algorithm as follows.

Step 2 (revised): Given $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$ and its corresponding transmission schedule S. Among all communication realizations with the number of receivers of each of their links being no more than κ, find the communication realization C that maximizes $\boldsymbol{\lambda}^T \boldsymbol{\gamma}(C)$. If $C \notin S$, add the communication realization and time proportion pair (C, δ_2) into S, where $\delta_2 > 0$ is the step size. If $C \in S$, increase its time proportion by δ_2. Then scale all the time proportions of communication realizations in S so that their sum equals 1.

Even though the revised Step 2 only involves a polynomial number of links, since a communication realization can simultaneously activate multiple links and the number of link combinations is exponential in the number of nodes, the complexity of the revised Step 2 is still exponential in the number of nodes. A simple but suboptimal approach to avoid such exponential complexity is to activate links sequentially to *construct* the communication realization C mentioned in the revised Step 2, as opposed to *search* it exhaustively. Unfortunately, we have to skip further discussions due to page limitations.

Note that Assumption 1 does not affect the validity of Proposition 2 as long as κ^* and $|I_{\min}(\boldsymbol{\gamma})|$ are also derived under Assumption 1.

C. Discussions

In both two steps of the revised algorithm, violating $\kappa \geq \kappa^*$ may result in a suboptimal solution.

Consider the network given in Example 1. If we let $\kappa = 1$, it is easy to see the algorithm will either find γ_1 or γ_2 as the minimum cut. The best communication realizations that maximize γ_1 and γ_2 are $C_1 = \{(e_{st_1}, 4)\}$ and $C_2 = \{(e_{st_2}, 4)\}$, respectively. Consequently, the iterative algorithm will converge to transmission schedule $S = \left\{\left(C_1, \frac{1}{2}\right), \left(C_2, \frac{1}{2}\right)\right\}$ with the corresponding multicast throughput being $R_{s\overline{t_1 t_2}} = 2$. This is suboptimal since $R_{s\overline{t_1 t_2}} = 3$ can be achieved by transmission schedule $\{(C_3, 1)\}$. The same example also shows that violating $\kappa \geq \kappa^*$ in Step 2 can lead to a suboptimal solution.

Since the value of κ^* is unknown before solving the optimization problem, a practical way to avoid requiring κ^* is to initialize κ with a small value and then, upon convergence of the algorithm, increase κ to check whether higher throughput can be achieved.

V. Finding The Minimum $s - T$ Cut

In the revised Step 1 of the iterative algorithm, given a transmission schedule with $s-T$ cut vector $\boldsymbol{\gamma}$, we need to find $I^{\kappa}_{\min}(\boldsymbol{\gamma})$, which is the indices of the first κ small-valued $s-T$ cuts. This needs to be done without exploring all $s-T$ cuts since the number of $s-T$ cuts is exponential in the number of nodes. The core of this problem is to find the minimum $s-T$ cut given a configuration graph. Due to the complication brought by hyperarc links, even if there is only one destination node t, the minimum $s-t$ cut can no longer be obtained using the well-known flow augmenting path algorithm [8].

Consider the network given in Example 2 whose configuration graph is illustrated in Figure 3. Consider the $s-t$ path consists of edges $e_{s\overline{ab}}$ and e_{at}. Assign the path with a flow of 1. It is easy to see there is no flow augmenting path (see definition in [8]). However, the achieved rate, $R_{st} = 1$ is not maximal since we can simultaneously assign flow 1 to path $(e_{s\overline{ab}}, e_{bt})$ and flow 1 to path (e_{sa}, e_{at}) to achieve $R_{st} = 2$. This shows that Corollary 5.2 of [8] does not hold for configuration graphs with hyperarc links.

Since given the transmission schedule the link rates are decoupled, the maximum flow can be derived using the algorithm proposed by Lun et al in [2], which also gives the minimum $s-T$ cut as a byproduct. Specifically, given the configuration graph $G(S) = (V, E)$ corresponding to transmission schedule S. Let g_{iJ} be the configuration rate of $e_{iJ} \in E$. Let s be the source node, $T = \{t_1, t_2, \ldots\}$ be the destination node set. We formulate the following optimization problem, with $x^{(t_k)}_{iJj}$, $\forall e_{iJ} \in E, j \in J, t_k \in T$, and R_{sT} being the variables.

$$\begin{aligned}
&\text{maximize } R_{sT} \\
&\text{s.t. } g_{iJ} \geq \sum_{j\in J} x^{(t_k)}_{iJj}, \quad \forall e_{iJ} \in E, t_k \in T \\
&\sum_{\{J|e_{iJ}\in E\}} \sum_{j\in J} x^{(t_k)}_{iJj} - \sum_{\{j|e_{jI}\in E, i\in I\}} x^{(t_k)}_{jIi} = \sigma^{(t_k)}_i, \\
&x^{(t_k)}_{iJj} \geq 0, \quad \forall e_{iJ} \in E, j \in J, t_k \in T
\end{aligned} \tag{12}$$

where

$$\sigma^{(t_k)}_i = \begin{cases} R_{sT} & \text{if } i = s \\ -R_{sT} & \text{if } i = t_k \in T \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

As shown in [2], (12) is a convex optimization problem, and hence can be solved efficiently with a polynomial complexity in the number of nodes and the number of links. The optimal R_{sT} is the maximum achievable throughput from s to T given the transmission schedule S. To achieve this multicast throughput, the actual flow over link e_{iJ} equals $\max_{t_k\in T} \sum_{j\in J} x^{(t_k)}_{iJj}$.

Consequently, we can construct an auxiliary configuration graph $\tilde{G}(S)$ from $G(S)$ by assigning $g_{iJ} - \max_{t_k\in T} \sum_{j\in J} x^{(t_k)}_{iJj}$ as the configuration rate to e_{iJ}. A link is removed if its configuration rate equals zero. Since R_{sT} equals the maximum multicast throughput, $\tilde{G}(S)$ must be disconnected. Suppose in $\tilde{G}(S)$, the node set V is divided into $K \geq 2$ disjoint subsets $\{V_0, V_1, V_2, \ldots, V_{K-1}\}$ each belongs to one connected subgraph, but there is no link connecting any of the two subgraphs. Assume V_0 contains the source node s and V_k, $k = 1, \ldots, K-1$ each contains at least one destination node. Recall that a cut of the graph is defined as a partition that divides the node set V into two disjoint subsets V_l, V_r. We can form a cut by assigning V_k, $k = 0, \ldots, K-1$, to V_l or V_r according to the following rules.

Minimum Cut Formulation:

- Assign V_0 to V_l, i.e., $V_l \supseteq V_0$.
- Choose an integer $1 \leq m \leq K-1$. Assign V_m to V_r.
- For all $1 \leq k \leq K-1$, $k \neq m$, assign V_k to either V_l or V_r.

Proposition 4: Any cut formed by the "Minimum Cut Formulation" is a minimum $s-T$ cut of graph $G(S)$.

Due to page limitations, we skip the demonstration that, based on the above minimum $s-T$ cut algorithm, we can also obtain the first κ small-valued $s-T$ cuts with a polynomial complexity in the number of nodes and the number of links.

References

[1] J. Wieselthier, G. Nguyen, and A. Ephremides, "Energy-efficient Broadcast and Multicast Trees in Wireless Networks", *Mobile Networks and Applications*, Vol. 7, pp. 481-492, 2002.

[2] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost Multicast over Coded Packet Networks", *IEEE Trans. Inform. Theory*, Vol. 52, pp. 2608-2623, Jun. 2006.

[3] R. Etkin, D. Tse, and H. Wang, "Gaussian Interference Channel Capacity to Within One Bit", submitted to *IEEE Transactions on Information Theory.*

[4] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow", *IEEE Trans. Inform. Theory*, Vol. 46, pp. 1204-1216, Jul. 2000.

[5] Y. Wu, M. Chiang, and S. Kung, "Distributed Utility Maximization for Network Coding Based Multicasting: A Critical Cut Approach", *Proc. 2nd Workshop Network Coding, Theory, Appl.*, Apr. 2006.

[6] J. Yuan, Z. Li, W. Yu, and B. Li, "A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks", *IEEE J. on Sel. A. in Commun.*, Vol. 24, pp. 2092-2103, Nov. 2006.

[7] Y. Sagduyu, A. Ephremides, "On Joint MAC and Network Coding in Wireless Ad Hoc Networks", *IEEE Trans. Inform. Theory*, Vol. 53, pp. 3697-3713, Oct. 2007.

[8] L. Ford and D. Fulkerson, "Flows in Networks", *Princeton University Press*, Princeton, NJ, 1962.