Forty-Sixth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 23-26, 2008

A Coded-Feedback Construction of Locally Minimum-Cost Multicast Network Codes

Chih-Chun Wang

Abstract— **There are two common network models for network coded traffic: one is the *fractional rate* model (the primary model for non-coded communication) and the other is the *integer rate model* used for detailed coding analysis on packet-by-packet behaviors. The existing approach of finding minimum-cost multicast network codes is based on the fractional rate model and solves the corresponding linear-programming (LP) problem. The LP-based network optimization generally converges slowly due to the small step size and does not take care of the packet-by-packet coding behavior of network coding.**

This paper develops a minimum-cost multicast scheme based on the integer rate model. The new scheme exploits a new concept of coded feedback, takes full advantages of the forward network-coded traffic, and possesses many practical advantages for efficient implementation. The complexity and performance of the coded-feedback scheme are studied both analytically and through simulations.

I. INTRODUCTION

For a unicast/mulitcast session, it is shown in [11] that linear network coding is capable of achieving the optimal min-cut max-flow rate. Many practical schemes have since been developed to realize the promised throughput gain for both wireline [2] and wireless networks [1]. The main ideas behind all practical schemes consist of using random linear network coding for its distributedness [9] plus tagging the coded content with the associated coding coefficients. To minimize the network usage of *a unicast session*, the user has to run the distributed max-flow algorithm to decide the max (s, d)-flow between source s and destination d, along which the coded packets will be sent [2].

There are at least two types of max-flow algorithms: one is the graph-theoretic algorithms such as the push-&-relabel (P&R) algorithm [7], [8], and the other is the linear-programming (LP) based network optimization. Being designed specifically as a graph-search algorithm, the former generally offers faster convergence speed and simpler implementation. The LP-based formulation is more complicated to implement and usually has slower convergence speed due to the small step sizes used in the gradient methods. With many complexity advantages, the graph-based algorithms also have their drawbacks. For example, it remains an open problem how to generalize the existing max (s, d)-flow algorithm when it is the *cost* instead of bandwidth being the objective function. On the other hand, the LP-based scheme can handle naturally different types of objective functions. The scheme proposed in this paper serves as a generalization to the graph-theoretic algorithms and possesses the same complexity advantages over LP-based solutions. We will show that with a new component of *coded feedback*, the graph-theoretic approach can handle various objective functions as do the LP-based algorithms. For the following, we focus on discussing and generalizing the graph-theoretic algorithms and the readers are referred to [13], [14] for more detailed discussion on LP-based solutions.

C.-C. Wang is with the Center for Wireless Systems and Applications (CWSA), School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA. chihw@purdue.edu

Existing max (s, d)-flow algorithms are based on the concept of *preflows*, which mirrors *the non-coded paradigm of network communication* but is incompatible to the packet-mixing nature of network coding. Therefore, the network generally has to run the preflow-based algorithm in parallel with the network-coded traffic, which significantly increases the control and communication overhead. The cost / bandwidth minimizing problem becomes even more challenging when considering multicast traffic instead of unicast traffic.

Consider a multicast session $(s, \{d_i\})$ problem, in which s is the source, each d_i is one destination, and $\{d_i\}$ is the set of destinations. Network coding shows strict throughput improvement over non-coded solutions in a multicast session. Nonetheless, to find bandwidth-efficient solutions, one currently has to run the distributed P&R max-flow algorithm for each (s, d_i) pair separately and then sends packets along *the union of the max (s, d_i)-flows.* The complexity thus grows linearly with respect to the number of destinations $|\{d_i\}|$. Moreover, *since the final network usage is the union of different max flows, a bandwidth-efficient solution requires as much edge-overlap between different max flows as possible in order to "save the overall bandwidth."* Nonetheless, the existing distributed P&R algorithm returns one max (s, d_i)-flow for each (s, d_i) pair at a time, and cannot control the edge-overlap between the max flows of different pairs. The union of the max flows identified by the P&R algorithm is thus far from minimal. For network coding in a multicast session, one needs a new max-flow algorithm that can *simultaneously and efficiently search for the max flows of different source-destination pairs* and can *minimize the total network usage of the union of all max (s, d_i)-flows.*

To that end, we develop a new coded-feedback scheme that starts from performing random linear network coding on a given network and then gradually trims the *redundant edges* until a bandwidth-minimal solution is achieved. Several other features of the proposed approach include

1) *Minimal control, communication, and complexity overhead:* All coded feedback packets are sent in the opposite direction of the forward data traffic and thus can be readily piggybacked to the acknowledgement packets

978-1-4244-2926-4/08/$25.00 ©2008 IEEE

with minimal control and communication overhead. Each coded feedback packet has a similar format as the header of the network coded forward data packet, namely, each coded feedback packet contains $|\{d_i\}|$ vectors of coding coefficients.

2) *No interruption to the forward traffic:* The proposed algorithm guarantees that the forward data traffic carries the same amount of useful information to the destination(s) throughout iterations. The proposed coded feedback algorithm is integrated seamlessly to the underlying network coding session with no interruption to the forward data traffic.
3) *Fast convergence speed:* The coded-feedback algorithm can identify multiple max flows simultaneously *within the same running time* as a traditional max-flow algorithm for the unicast traffic. The only expense is the size of each coded feedback packet, which grows linearly with respect to the number of destinations $|\{d_i\}|$.
4) *Adaptability to different design criteria:* The coded feedback algorithm works for arbitrary size of the finite field $\mathsf{GF}(q)$ used in network coding, and can be easily adapted to finding max flows with short delay and low power consumption.

The first coded-feedback algorithm was introduced in our preliminary work [12], which is designed specifically for unicast traffic, cannot be generalized for multicast traffic, and does not admit the same level of distributiveness as our new algorithm. Other related works include using network coded packets to explore unknown networks for network tomography and monitoring [6], [3], [4].

The remainder of this paper is organized as follows. Section II discusses the integer-rate network model. Section III introduces our main contribution, a new coded feedback algorithm that applies to both unicast and multicast traffic for the first time. The proofs of the correctness of the proposed algorithm is sketched in Section IV. Section V discusses the low-complexity distributed implementations for multicast sessions with delay constraints and for minimizing power consumption. Numerical experiments are reported in Section VI. Section VII concludes this paper.

II. THE INTEGER-RATE NETWORK MODEL

We consider only directed acyclic graphs (DAGs) $G = (V, E)$ where each edge is able to carry one packet per unit time, say a second. The propagation delay is also one second. High capacity links are modelled by parallel edges and long delay links are modelled by long paths.

We use $\mathrm{In}(v), \mathrm{Out}(v) \subseteq E$ to denote all edges entering or leaving node v respectively. We consider either a single unicast session from source s to destination d or a single multicast session from source s to destinations $\mathbf{d} = \{d_i\}$. Without loss of generality, we assume that $\mathrm{In}(s) = \emptyset$ and $\mathrm{Out}(d) = \emptyset$ (or $\mathrm{Out}(d_i) = \emptyset$ for all $d_i \in \mathbf{d}$), and we assume that $\mathrm{In}(v) \neq \emptyset$ and $\mathrm{Out}(v) \neq \emptyset$ for all $v \notin \{s, d\}$ (or $v \notin \{s\} \cup \{d_i\}$).

III. A NEW CODED FEEDBACK ALGORITHM

Discussion in this section focuses on a single unicast session. The generalization for a single multicast session is relegated to Section V.

A. The Basic Structure

We consider the following *time-invariant network coding scheme*. Source s sends continuously packets $\{X_{i,t}\}_{i=1,\cdots,|\mathrm{Out}(s)|}$ along edges in $\mathrm{Out}(s)$ to its neighbors $\{w_i\}_{i=1,\cdots,|\mathrm{Out}(s)|}$ in the t-th second. In the practical network coding scheme by [2], the incoming packets are buffered and packet mixing is allowed among packets of the same time stamp (also known as "generations"). Denote $n \triangleq |\mathrm{Out}(s)|$. The coding vector for each edge $e = (u, v)$, denoted by m_e or $m_{u,v}$, is an n-dimensional *row vector* in $\mathsf{GF}(q)$. The i-th component corresponds to the coefficient with respect to $X_{i,t}$. For example, m_{s,w_i} is a delta vector δ_i with all but the i-th component being zero and the i-th component being one. For any $E' \subseteq E$, since each message m is an n-dimensional row vector, we define the following bracket notation $[m_e : e \in E']$ as an $|E'| \times n$ matrix constructed by vertically concatenating all m_e row vectors on E'. We use $\langle m_e : e \in E' \rangle$ to denote the row space of $[m_e : e \in E']$.

Before computing the coding vectors m along the network, we perform the following initialization.

§ INITIALIZATION

1: Each edge $(s, w_i) \in \mathrm{Out}(s)$ has m_{s,w_i} set to a δ_i vector.
2: Each node $v \in V \backslash \{s, d\}$ chooses arbitrarily an $|\mathrm{Out}(v)| \times |\mathrm{In}(v)|$ coefficient matrix $\Gamma(v) = [\gamma_{w,u}(v)]$ where each entry $\gamma_{w,u}(v) \in \mathsf{GF}(q)$ for all $(v, w) \in \mathrm{Out}(v)$ and $(u, v) \in \mathrm{In}(v)$.

Matrix $\Gamma(v) = [\gamma_{w,u}(v)]$ is the mixing/transfer coefficients within node v, which will clear in the description of the next sub-routine. The same $\Gamma(v)$ will be used for packets of different generations, and this scheme is thus termed *time-invariant* network coding.

§ COMPUTE FORWARD MESSAGES

Every second, each node $v \in V \backslash d$ performs the following tasks until all forward messages m_e in the network are *stabilized* and do not change anymore.

1: $[m_{v,w} : (v, w) \in \mathrm{Out}(v)] \leftarrow \Gamma(v) \cdot [m_{u,v} : (u, v) \in \mathrm{In}(v)]$.

COMPUTE FORWARD MESSAGES is simply an algorithmic way of describing packet mixing and broadcasting in a practical network coding scheme [2].

The first novel component of the coded feedback algorithm is the introduction of *the coded feedback message* $q_{u,v}$, which is an n-dimensional row vector sent in the opposite direction of $m_{u,v}$. Namely, for any $(u, v) \in \mathrm{In}(v)$, its coded feedback $q_{u,v}$ is a linear combination of $\{q_{v,w} : (v, w) \in \mathrm{Out}(v)\}$. We use the following sub-routine to compute the feedback

messages q, in which we use $\mathrm{Rank}(d) \triangleq \mathrm{Rank}([m_e : e \in \mathrm{In}(d)])$ as shorthand.

§ COMPUTE CODED FEEDBACK

1: Destination d *randomly* constructs $(n-\mathrm{Rank}(d))$ vectors $\{m_a^* : a \in \{1, \cdots, n-\mathrm{Rank}(d)\}\}$ such that jointly $\{m_e : e \in \mathrm{In}(d)\}$ and $\{m_a^*\}$ span the entire n-dimensional vector space.

2: Destination d *randomly*[1] constructs two sets of vectors $\{q_e : e \in \mathrm{In}(d)\}$ and $\{q_a^* : a \in \{1, \cdots, n-\mathrm{Rank}(d)\}\}$ such that the following matrices

$$\mathbf{q} \triangleq \begin{pmatrix} [q_e : e \in \mathrm{In}(d)] \\ [q_a^* : a \in \{1, \cdots, n-\mathrm{Rank}(d)\}] \end{pmatrix}$$
$$\mathbf{m} \triangleq \begin{pmatrix} [m_e : e \in \mathrm{In}(d)] \\ [m_a^* : a \in \{1, \cdots, n-\mathrm{Rank}(d)\}] \end{pmatrix}$$

satisfy $\mathbf{q}^{\mathrm{T}}\mathbf{m} = \mathbf{I}_n$ where $\mathbf{I}_n$ is an n by n identity matrix and $\mathbf{q}^{\mathrm{T}}$ is the transpose of $\mathbf{q}$.

3: Destination d sends out the newly constructed $\{q_e : e \in \mathrm{In}(d)\}$ along each edge in $\mathrm{In}(d)$.

4: Every second, each node $v \in V\backslash v$ computes q_e by $[q_e : e \in \mathrm{In}(v)] \leftarrow \Gamma(v)^{\mathrm{T}}[q_e : e \in \mathrm{Out}(v)]$ until all feedback vectors q_e in the network are *stabilized* and do not change anymore.

Note that both the forward and backward messages share the same coefficient matrix $\Gamma(v)$. The only difference is that for q_e, the input/output roles have been reversed and we use the transpose $\Gamma(v)^{\mathrm{T}}$ as the feedback mixing matrix. In practice, q_e can be easily piggybacked to the acknowledgement packet.

The basic structure of a coded feedback algorithm can then be described as follows.

§ BASIC CODED FEEDBACK ALGORITHM

1: INITIALIZATION
2: **loop**
3: COMPUTE FORWARD MESSAGES
4: COMPUTE CODED FEEDBACK
5: Let $E_R(v)$ denote the output of IDENTIFY REDUNDANT EDGES, where $E_R(v) \subseteq \mathrm{In}(v)$ for some node $v \in V\backslash s$.
6: **if** $E_R(v) \neq \emptyset$ **then**
7: Remove $E_R(v)$.
8: **else**
9: **return** the remaining graph G
10: **end if**
11: **end loop**

[1]As in a typical random network algorithms, the randomness in Lines 1 and 2 is needed for computing min-cut correctly with high probability. (Refer to Proposition **??** below.) The detailed "random constructions" used in Lines 1 and 2 are described briefly as follows. For Line 1, d arbitrarily chooses the complementing vectors $[m_a^*]$. Then, d randomly selects an $(n - \mathrm{Rank}(d)) \times |\mathrm{In}(d)|$ matrix Γ_d with each entry uniformly and independently distributed. Replace $[m_a^*]$ by $[m_a^*] + \Gamma_d[m_e : e \in \mathrm{In}(d)]$. In this way, the new $[m_a^*]$ is randomly constructed.

For Line 2, d first identifies $\mathrm{Rank}(d)$ independent m_e in $[m_e : e \in \mathrm{In}(d)]$. Without loss of generality, assume that it is the last $\mathrm{Rank}(d)$ rows. Then the $\mathbf{m}$ matrix can be decomposed as an $(|\mathrm{In}(d)| - \mathrm{Rank}(d)) \times n$ matrix $\mathbf{m}_1$ and an $n \times n$ invertible matrix $\mathbf{m}_2$. We then construct two corresponding sub-matrices $\mathbf{q}_1$ and $\mathbf{q}_2$ as follows. Destination d first randomly chooses a $\mathbf{q}_1$ with each entry uniformly and independently distributed. Then, it finds the unique $\mathbf{q}_2$ satisfying $\mathbf{q}_2^{\mathrm{T}} = (\mathbf{I}_n - \mathbf{q}_1^{\mathrm{T}}\mathbf{m}_1)\mathbf{m}_2^{-1}$. The newly found $\mathbf{q}_1$ and $\mathbf{q}_2$ are concatenated to form the desired $\mathbf{q}$. The randomly chosen $\mathbf{q}_1$ will ensure the $\mathbf{q}_2$ and the entire $\mathbf{q}$ matrix being randomly constructed.

The subroutine IDENTIFY REDUNDANT EDGES takes the stabilized forward and feedback messages m_e and q_e as input, and outputs an edge set $E_R(v) \subseteq \mathrm{In}(v)$ for some $v \in V\backslash s$. The goal is to design a distributed subroutine IDENTIFY REDUNDANT EDGES such that its output $E_R(v)$ *contains only edges that can be safely removed without affecting the dimension of the space received by* d. The first of such IDENTIFY REDUNDANT EDGES was introduced in [12] and this paper develops a new, more flexible IDENTIFY REDUNDANT EDGES based on a different philosophy.

B. A New Subroutine IDENTIFY REDUNDANT EDGES

In this subsection, we provide an ALGEBRA-BASED IDENTIFY REDUNDANT EDGES (AB-IRE) that computes the *redundant edges* $E_R(v)$ *in a distributed fashion.*

§ ALGEBRA-BASED IDENTIFY REDUNDANT EDGES

1: **if** there exist a $v \in V$ and a non-empty $\Xi \subseteq \mathrm{In}(v)$ satisfying the following properties: (i) let Π denote the $|\Xi| \times |\Xi|$ square matrix $\Pi \triangleq [q_e : e \in \Xi] \cdot [m_e : e \in \Xi]^{\mathrm{T}}$, and (ii) $I_{|\Xi|} - \Pi$ is of full rank **then**
2: $E_R(v)$ can be chosen arbitrarily as any of such Ξ.
3: **else**
4: $E_R(v) \leftarrow \emptyset$
5: **end if**

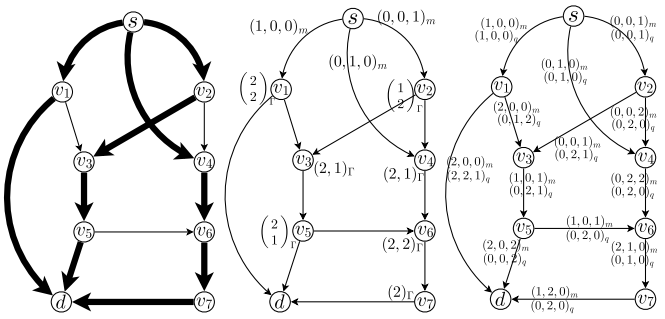
The new subroutine AB-IRE is very different from the first IDENTIFY REDUNDANT EDGES in [12]. The most distinct feature of AB-IRE is that AB-IRE can search the $\mathrm{In}(v)$ of any node v in any arbitrary order. For comparison, the existing construction in [12] searches the redundant edge set $E_R(v)$ from the most downstream node $v = d$ and back to the most upstream node $v = s$. This key feature of our new scheme enables us to construct a locally minimum-cost multicast network code as will be described in Section V-D.

Note that it is computationally expensive to compute the rank of $I_{|\Xi|} - \Pi$ for Ξ of large dimension. Nonetheless, one can limit our choices of Ξ to be those containing a single edge. For those Ξ, one simply needs to check whether $q_e m_e^{\mathrm{T}} = 1$, which can be achieved ultra efficiently.

C. One Illustrative Example

Consider a directed acyclic network as illustrated in Fig. 1(a). Source s would like to transmit at the optimal rate to destination d using a network with 7 other intermediate nodes. One max flow between s and d is illustrated by thick arrows, and the corresponding max-flow value is 3.

Consider a small finite field $\mathsf{GF}(3)$. The mixing matrices $\Gamma(v)$ chosen in INITIALIZATION subroutine is illustrated in Fig. 1(b). We add the subscript Γ to emphasize it is a transfer



(a) The Network (b) The mixing matrices used to compute m_e and q_e (c) Compute the forward and feedback coding vectors

Fig. 1. An illustrative example of the BASIC CODED FEEDBACK ALGORITHM: (a) The underlying network, for which the max flow is illustrated by thick arrows, (b) After the INITIALIZATION step, (c) After the COMPUTE FORWARD MESSAGES(0) step, (d) After the COMPUTE FORWARD MESSAGES(1) step, and (e) After the COMPUTE CODED FEEDBACK step.

matrix rather than a vector. Source s sends the coding vectors $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ along edges (s,v_1), (s,v_4), and (s,v_2). The forward and feedback coding vectors can then be computed as in Fig. 1(c). Note that to distinguish the forward messages m_e and the coded feedback q_e, we add subscript m or q to each row vector in Fig. 1(b). For example, $\Gamma(v_2) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, which means $m_{v_2,v_3} = 1 \cdot m_{s,v_2}$ and $m_{v_2,v_4} = 2 \cdot m_{s,v_2}$. Similarly, $\Gamma(v_3) = (2,1)$ means $m_{v_3,v_5} = 2m_{v_1,v_3} + m_{v_2,v_3}$.

The initial coded feedback vectors along $\mathrm{In}(d)$ are computed by solving the following matrix equation.

$$\begin{bmatrix} q_{v_1,d} \\ q_{v_5,d} \\ q_{v_7,d} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} m_{v_1,d} \\ m_{v_5,d} \\ m_{v_7,d} \end{bmatrix} = I_3 \Leftrightarrow \begin{bmatrix} q_{v_1,d} \\ q_{v_5,d} \\ q_{v_7,d} \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix}.$$

Other q_e are computed by the transpose of $\Gamma(v)$. For example, since $\Gamma(v_7)$ of the new graph is 2, we have $q_{v_6,v_7} = 2q_{v_7,d} = (0,1,0)$ in Fig. 1(c). Since $\Gamma(v_6) = (2,2)$, we have $q_{v_5,v_6} = q_{v_4,v_6} = 2q_{v_6,v_7} = (0,2,0)$. Since $\Gamma(v_5) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, we have $q_{v_3,v_5} = 2q_{v_5,d} + q_{v_5,v_6} = (0,2,1)$.

In AB-IRE, nodes can be searched in any order. Suppose the first node to search is v_3. There are three possible collections of redundant edges: $\Xi_1 = \{(v_1,v_3)\}$, $\Xi_2 = \{(v_2,v_3)\}$, and $\Xi_3 = \{(v_1,v_3),(v_2,v_3)\}$. The corresponding $I - \Pi$ matrices are

$$1 - \Pi_1 = 1 - q_{v_1,v_3} m_{v_1,v_3}^{\mathrm{T}} = 1 - 0 = 1$$
$$1 - \Pi_2 = 1 - q_{v_2,v_3} m_{v_2,v_3}^{\mathrm{T}} = 1 - 1 = 0$$
$$I_2 - \Pi_3 = I_2 - \begin{bmatrix} q_{v_1,v_3} \\ q_{v_2,v_3} \end{bmatrix} \begin{bmatrix} m_{v_1,v_3} \\ m_{v_2,v_3} \end{bmatrix}^{\mathrm{T}}$$
$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

As can be easily checked, the only full rank choice is $1 - \Pi_1$. Therefore, $\Xi_1 = \{(v_1,v_3)\}$ is a redundant edge set. For comparison, a max-flow of the network is computed offline and illustrated in Fig. 1(a). Edge (v_1,v_3) is indeed a redundant edge not participating in the max-flow.

IV. THE CORRECTNESS OF THE PROPOSED ALGORITHM

A. Quantifying The Benefits of The New Algorithm

In this subsection, we will quantify the benefit of the coded feedback algorithm when the subroutine AB-IRE is used. The following properties hold for any choice of Ξ made in AB-IRE.

Proposition 1: For any finite field size $\mathsf{GF}(q)$, the following properties hold for the coded feedback algorithm using the AB-IRE subroutine.

1) The BASIC CODED FEEDBACK ALGORITHM stops in $2l|E|$ seconds where l is the length of the longest path in the network;
2) Throughout the main loop in Line 2 of the BASIC CODED FEEDBACK ALGORITHM, the dimension of $\langle m_{u,d} : (u,d) \in \mathrm{In}(d) \rangle$ remains unchanged;
3) Suppose the row rank of $\langle m_e : e \in \mathrm{In}(d) \rangle$ equals n, the number of non-coded symbols n. Consider the final output graph of the coded feedback algorithm.[2] If the coefficients $\Gamma(\cdot)$ cannot be changed, then removing any non-empty subset of $\mathrm{In}(v)$ of any given v will decrease strictly the rank of the space received by d.

We have to emphasize that Property 3 of Proposition 1 does not exclude the case in which removing *simultaneously* two non-empty subsets of $\mathrm{In}(v_1)$ and $\mathrm{In}(v_2)$ for a given pair of distinct nodes $v_1 \neq v_2$ might still keep the dimension of the received space unchanged. However, if each node v has to make its own decision *locally* whether to remove a subset of $\mathrm{In}(v)$, then no further edge reduction can be made. Accordingly, we say the BASIC CODED FEEDBACK ALGORITHM with AB-IRE finds a network coding solution with *locally minimal network usage*.

To further bridge the above algebraic properties and the graph-theoretic definition of max-flows, we rely on the notions of a sufficiently large $\mathsf{GF}(q)$ and random linear coding in [9]:

Proposition 2: Suppose the $\mathsf{GF}(q)$ size is sufficiently large. Then the following two properties are satisfied with close-to-one probability.

1) The output of the coded feedback algorithm is a max (s,d)-flow;

[2]The additional constraint that the row rank of $\langle m_{u,d} : (u,d) \in \mathrm{In}(d) \rangle$ being n is unique for Property 3 while Property 2 holds regardless of the dimension of $\langle m_{u,d} : (u,d) \in \mathrm{In}(d) \rangle$. Since the goal of any practical network coding solution is to recover *all* n *symbols* in a given generation, this additional condition imposes little restriction for practical applications. In practice, when the achievable max-flow-value rate is less than n, source s sends *linearly coded (dependent) packets* to facilitate decoding at d instead of sending linearly independent packets. The parity-check constraints of the coded packets are common knowledge shared by both s and d so that d can decode the original information. Therefore, sending coded packets can be modelled as sending independent packets plus allowing auxiliary "virtual pipes" to carry the parity-check information directly from s to d. After this conversion, the "augmented network coding solution" has the rank of $\langle m_{u,d} : (u,d) \in \mathrm{In}(d) \rangle$ being n. The additional constraint is satisfied.

2) The coded feedback algorithm stops in $2l|V|$ seconds if for any v, we choose a maximal $\Xi \subseteq \mathrm{In}(v)$ in Line 2 of AB-IRE and output the maximal Ξ as $E_R(v)$.

Proposition 2 ensures that when a generic LCM (or random network coding) is used, the running time of the coded feedback algorithm is $\mathcal{O}(|V|^2)$, which is the same as the traditional distributed P&R max-flow algorithms.

B. Proof of Proposition 1

Due to the lack of space, we provide in details only the following central lemma for proving the correctness of the coded feedback algorithm with AB-IRE, which is a strengthened version of the Sylvester's determinant theorem for the finite field. A brief sketch of how to use the following lemma for proving the correctness is also provided.

Consider two $m \times n$ matrices A and B in $\mathsf{GF}(q)$ where $m \leq n$.

Lemma 1: For any $k \times k$ square matrix X, we define the *rank deficiency* of X being $k - \mathrm{Rank}(X)$. Then the two square matrices

$$I_m - AB^{\mathrm{T}} \text{ and } I_n - B^{\mathrm{T}}A \tag{1}$$

have the same rank deficiency.

Proof: Case 1: We first assume that A is of full row rank. (This assumption will be relaxed later.) Without loss of generality, we can also assume the first m columns of A is a full rank square matrix by reordering the column indices of A and B accordingly. We then represent A by $A = [A_m \ A_{n-m}]$. Similarly, represent B by $B = [B_m \ B_{n-m}]$. We first left multiply $(I_n - A^{\mathrm{T}}B)$ by the following matrix:

$$\begin{aligned}
&\begin{bmatrix} I_m & 0 \\ -A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & I_{n-m} \end{bmatrix}(I_n - A^{\mathrm{T}}B) \\
&= \begin{bmatrix} I_m & 0 \\ -A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & I_{n-m} \end{bmatrix} \\
&\qquad \cdot \left(I_n - \begin{bmatrix} A_m^{\mathrm{T}} \\ A_{n-m}^{\mathrm{T}} \end{bmatrix}[B_m \ B_{n-m}]\right) \\
&= \begin{bmatrix} I_m & 0 \\ -A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & I_{n-m} \end{bmatrix} - \begin{bmatrix} A_m^{\mathrm{T}}B_m & A_m^{\mathrm{T}}B_{n-m} \\ 0 & 0 \end{bmatrix}.
\end{aligned} \tag{2}$$

We continue left-multiplying (2) by the following matrix.

$$\begin{aligned}
&\begin{bmatrix} I_m & A_m^{\mathrm{T}}B_{n-m} \\ 0 & I_{n-m} \end{bmatrix} \cdot (2) \\
&= \begin{bmatrix} I_m - A_m^{\mathrm{T}}B_{n-m}A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & A_m^{\mathrm{T}}B_{n-m} \\ -A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & I_{n-m} \end{bmatrix} \\
&\qquad - \begin{bmatrix} A_m^{\mathrm{T}}B_m & A_m^{\mathrm{T}}B_{n-m} \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} I_m - A_m^{\mathrm{T}}B_{n-m}A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)\mathrm{T} - A_m^{\mathrm{T}}B_m & 0 \\ -A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} & I_{n-m} \end{bmatrix}.
\end{aligned} \tag{3}$$

We note that in the above two equalities, we only multiply $I_n - A^{\mathrm{T}}B$ by matrices of full rank. Therefore, the rank deficiency of $I_n - A^{\mathrm{T}}B$ is the same as the rank deficiency of (3), which is also the same as the rank deficiency of the upper right submatrix of (3):

$$\begin{aligned}
&I_m - A_m^{\mathrm{T}}B_{n-m}A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} - A_m^{\mathrm{T}}B_m \\
&= A_m^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} - A_m^{\mathrm{T}}B_{n-m}A_{n-m}^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} \\
&\qquad - A_m^{\mathrm{T}}B_mA_m^{\mathrm{T}}\left(A_m^{-1}\right)^{\mathrm{T}} \\
&= A_m^{\mathrm{T}}\left(I_m - [B_m \ B_{n-m}]\begin{bmatrix} A_m^{\mathrm{T}} \\ A_{n-m}^{\mathrm{T}} \end{bmatrix}\right)\left(A_m^{-1}\right)^{\mathrm{T}}.
\end{aligned}$$

Since A_m is of full rank, the rank deficiency of the above matrix is the same as the rank deficiency of $I_m - BA^{\mathrm{T}}$. The proof is thus complete for the case in which A is of full rank.

Case 2: A is of row rank $m' < m$. We can rewrite the $m \times n$ matrix A as the product of an $m \times m'$ matrix Δ and an $m' \times n$ matrix A' such that $A = \Delta A'$. Furthermore, Δ is of full column rank m' and A' is of full row rank m'.

We then have

$$I_n - B^{\mathrm{T}}A = I_n - B^{\mathrm{T}}(\Delta A') = I_n - (B^{\mathrm{T}}\Delta)A'. \tag{4}$$

Since A' is of full row rank, by the proof of Case 1, the rank deficiency of (4) equals the rank deficiency of

$$I_{m'} - A'(B^{\mathrm{T}}\Delta) = I_{m'} - (A'B^{\mathrm{T}})\Delta. \tag{5}$$

Since Δ is of full column rank, by a similar proof of Case 1, the rank deficiency of (5) equals the rank deficiency of

$$I_m - \Delta(A'B^{\mathrm{T}}) = I_m - (\Delta A')B^{\mathrm{T}} = I_m - AB^{\mathrm{T}}.$$

The proof is complete. ■

The sketch of the proof of Proposition 1: The computation time analysis of the coded feedback algorithm with AB-IRE is straightforward. Since each iteration of the main loop in the BASIC CODED FEEDBACK ALGORITHM will remove at least one edge, there are at most $|E|$ iterations before termination. Each iteration takes at most $2l$ seconds for both the forward and feedback messages to stabilize. The entire algorithm thus stops within $2l|E|$ seconds.

Property 2 relies heavily on Lemma 1. Since the construction of AB-IRE ensures that $I_{|\Xi|} - \Pi$ is of full rank, Lemma 1 ensures that

$$I_n - [q_e : e \in \Xi]^{\mathrm{T}}[m_e : e \in \Xi] \tag{6}$$

is also of full rank. One can show that the above matrix being of full rank is a sufficient condition that after deleting Ξ, the rank of $[m_e : e \in \mathrm{In}(d)]$ will not decrease.

During the proof of Property 3, one can show that when the received rank of $[m_e : e \in \mathrm{In}(d)]$ being n, the sufficient condition (6) being of full rank is also a necessary condition for the statement that deleting Ξ, the rank of $[m_e : e \in \mathrm{In}(d)]$ will not decrease. As a result, once the coded feedback algorithm stops, i.e. no other Ξ satisfying $I_{|\Xi|} - \Pi$ being of full rank, it is guaranteed that removing any non-empty $E_R(v)$ will strictly decrease the received rank. The proof is thus complete. ■

V. IMPLEMENTATIONS & GENERALIZATIONS

A. Low-Complexity Distributed Implementation

The BASIC CODED FEEDBACK ALGORITHM can be made distributed in a straightforward manner. The subroutines COMPUTE FORWARD MESSAGES(b) and COMPUTE CODED FEEDBACK can be easily implemented in a network as it is a direct analogy of the network coding traffic. Deciding which node v and the associated $E_R(v) \subseteq \text{In}(v)$ to remove by IDENTIFY REDUNDANT EDGES is the only subroutine for which some form of coordination is necessary. This coordination can be achieved by a token-based approach. Each intermediate node calculates its own Π matriix and checks whether it satisfies the criteria in AB-IRE. Consider those v that have non-empty $E_R(v)$ subject to edge deletion. Each v requests a token from the destination d. Using a single token ensures that one and only one v is allowed to remove $E_R(v)$, or equivalently to stop its incoming traffic. Passing the token along the data / acknowledgement packets also enforces sufficiently long waiting time for m_e and q_e to stabilize.

An even more decoupled scheme is that each v that has a non-empty $E_R(v)$ just waits a random number of seconds before stopping the traffic on $E_R(v)$. A carefully designed random waiting time (with carefully designed reset mechanisms) ensures that in high probability, no two nodes trim the graph simultaneously and the waiting time for m_e and q_e to stabilize is sufficient. Even when two nodes perform graph-trimming in too short a time period, the rank of $\langle m_e : e \in \text{In}(d)\rangle$ decreases only if the pruned edges happen to be in a minimum cut, a rare event in a random network [10]. In summary, a small probability that the rank may decrease slightly is a welcome compromise for a perfectly distributed algorithm.

B. Searching Max Flows with Constraints on Delay or Coding Complexity

The coded feedback algorithm focuses on maintaining the same dimension of the received space while trimming the network usage. Therefore, even in the (relatively infrequent) cases in which random linear network coding achieves only a near-optimal rate r that is strictly less than the max-flow-value, one can still apply the coded feedback algorithm to prune the unnecessary edges while sustaining the near-optimal rate r. This is a highly desired feature for practical systems. For example, with the delay and complexity requirements, a user is likely to be interested only in (s, d) paths that use fewer than h hops. Then the user can use a *controlled broadcast* plus network coding scheme that explores only paths of length $< h$. The optimal max-flow-value rate may not be attainable under this partial network exploration. The coded feedback algorithm allows the user to still achieve the best possible rate r and prune the redundant traffic.

Similarly, the complexity constraint may also limit how large the $\mathsf{GF}(q)$ size that can be employed in the network. In some cases, the allowed $\mathsf{GF}(q)$ size is too small and one cannot achieve the max-flow-value rate in the network (especially when a multicast session is considered) [5]. Since the coded feedback algorithm applies to arbitrary size of $\mathsf{GF}(q)$, one can still maintain the best achievable rate while trimming the network usage in the optimal way.

C. Solving Multiple Max Flows Simultaneously

The coded feedback algorithm with AB-IRE can be generalized for the multicast session problem so that the output is a multicast network coding solution with *locally minimal network usage*.

Consider the same setting of directed acyclic graphs with unit edge capacity and unit edge delay. Instead of a unicast session from source s to destination d, we have a multicast session from source s to destinations $\mathbf{d} = \{d_i\}$. A generalized coded feedback algorithm is described as follows.

§ GENERALIZED CODED FEEDBACK ALGORITHM FOR THE MULTICAST PROBLEM

INITIALIZATION
loop
  COMPUTE FORWARD MESSAGES
  COMPUTE CODED FEEDBACK for each destination d_i respectively. Namely, each edge e carries $|\mathbf{d}|$ coded feedback messages $\{q_{i,e} : \forall d_i \in \mathbf{d}\}$.
  Let $E_R(v)$ denote the output of AB-MULTI-RE, where $E_R(v) \subseteq \text{In}(v)$ for some node $v \in V \backslash s$.
  if $E_R(v) \neq \emptyset$ **then**
    Remove $E_R(v)$.
  else
    return the remaining graph G
  end if
end loop

In the above algorithm, we use the following subroutine:

§ ALGEBRA-BASED IDENTIFY MULTI-REDUNDANT EDGES (AB-MULTI-RE)

if there exist a $v \in V$ and a non-empty edge subset $\Xi \subseteq \text{In}(v)$ satisfying the following properties for all $d_i \in \mathbf{d}$: (i) let Π_i denote the $|\Xi| \times |\Xi|$ square matrix $\Pi_i = [q_{i,e} : e \in \Xi] \cdot [m_e : e \in \Xi]^{\mathrm{T}}$, and (ii) $I_{|\Xi|} - \Pi_i$ is of full rank **then**
  $E_R(v)$ can be chosen arbitrarily as any of such Ξ.
else
  $E_R(v) \leftarrow \emptyset$
end if

The above algorithm and the corresponding subroutine are straightforward generalizations for BASIC CODED FEEDBACK ALGORITHM and AB-IRE. *All existing properties in Propositions 1 and 2 for the basic coded feedback algorithm hold for the generalized coded feedback algorithm as well.*

D. Greedy Search for Max Flows With Minimal Cost

All the algorithms discussed previously are generic as they allow flexible designs for the performance and complexity tradeoff. For example, in AB-IRE and in AB-MULTI-RE,

one has the freedom of designing an arbitrary search order of v for which the redundant edge set $E_R(v)$ will be returned. Even for the same given v, there might be more than two choices of Ξ. Searching for those Ξ containing more edges accelerates the convergence as one could remove more edges in a single round. However, searching for large Ξ requires more computational resources. One can thus strike a balance of the computation resources within a single node and the convergence speed of the entire network.

For the following, we demonstrate the flexibility of the proposed algorithm by using the generalized coded feedback algorithm as a greedy algorithm searching for the union of max flows while minimizing the total cost.

Consider the single source multicast problem from s to $\mathbf{d}$. In addition to the setting discussed previously, each edge e charges a non-uniform price $c(e)$ for carrying one packet. Our goal is to find a subgraph G' such that the max-flow values between all (s, d_i) in G' are identical to those in the original graph G, while at the same time the total cost $\sum_{e\in G'} c(e)$ is minimized. The cost function $c(e)$ represents generally the power consumption for a given link. A reasonable choice is thus

$$c(e) = \frac{1}{\text{multiplicity of edge } e} \tag{7}$$

where the multiplicity of edge $e = (u, v)$ is the number of parallel edges connecting nodes u and v. To be more specific, in a given (wireless) network, each node can use certain power level to send packets along a given link. Depending on the noise level of the given link, different data rates can be achieved for different links (even when the same power level is used). Since links with high capacity are modelled by multiple parallel edges, the power (or time) consumption for each edge in a given link is thus (7). Minimizing the total power thus corresponds to minimizing $\sum_{e\in G'} c(e)$.

The generalized algorithm can be easily modified as a greedy search algorithm for the above problem. To that end, random linear network coding is used in INITIALIZATION, which ensures a generic LCM is generated with a high probability. Since the output $E_R(v)$ of AB-MULTI-RE can be chosen arbitrarily as any valid Ξ, the greedy search simply returns an edge set $E_R(v)$ that is the valid Ξ having highest cost per edge. By sequentially removing the redundant edge sets with the highest average cost, the generalized coded feedback algorithm acts as a greedy algorithm and reaches a locally optimal solution.

It is worth noting that the proposed greedy algorithm focuses on an intrinsically hard, integer programming problem as each e is either within G' or outside G'. Therefore the above greedy approach does not guarantee a globally optimal solution. Only the local optimality is guaranteed. For comparison, the LP-based formulations [13], [14] focus on the fractional-rate setting, which corresponds to solving relatively easier linear programming problems at the cost of more complicated implementation and slower convergence time. The integral setting considered in this work takes into account jointly packet-by-packet coding behavior and rate-control for the first time in the literature.

VI. NUMERICAL EXPERIMENTS

In this section, numerical experiments are conducted for the generalized coded feedback algorithm that greedily searches for the minimum cost solution as described in Section V-D.

We use the generic P&R algorithm [8] as benchmark, which also admits distributed network implementation. For numeric comparisons, we consider a randomly generated, sparse, 30-node DAG, of which each node is connected to 5.2 links in average. The capacity of each link (or equivalently the multiplicity of each edge) is randomly chosen from 1 to 10. To describe explicitly the network of interest, we provide its 30×30 incidence matrix $A = \begin{bmatrix} A_{1-15} \\ A_{16-30} \end{bmatrix}$, where each entry $A_{i,j}$ of A is the number of parallel directed edges connecting nodes i and j. If we use "a" as the hexadecimal digit for 10 and use "." for zeros, the two 15×30 sub-matrices A_{1-15} and A_{16-30} can be expressed as follows.

```
........a8.....................  ..................6...3a.a.....
...a....3.....................  ....................a8a.......
...44..121.....................  ......................6.68.....
.....a....3.....................  ......................6.395...
.....59...a.8...................  ....................65..48..9.
.........3....8.................  ......................1...3.9.a
.........16.5...8...............  .......................6....1.
.........4a....82...............  .......................6381....
..........8.6.1.................  ........................1..8.
............9..7................  .............................4.
.............5..2a3..6..........  ..............................8
................6.2.............  ..............................
................a..a.7..........  ..............................
.................5a.............  ..............................
.................4....56.......,  ..............................
```

The cost for each edge is assigned according to (7), which corresponds to the total power/time consumption of the network coding solution. In each round, AB-MULTI-RE selects the $E_R(v) \subseteq \text{In}(v)$ that has the largest average cost per edge, which will be removed from the existing graph. The source $s = 1$ and we choose the destination set $\mathbf{d}$ to be $\{28, 29, 30\}$.

There are three destinations in our numerical experiment. The P&R algorithm needs to find three max flows between each source-destination pair before taking the union of all three max flows. Finding the max $(1, 28)$, $(1, 29)$, $(1, 30)$-flows takes 374, 43, and 487 seconds respectively. The total convergence time for finding three max flows is thus $487 + 43 + 374 = 904$ seconds. We use $G_{\text{P\&R},28}$, $G_{\text{P\&R},29}$, and $G_{\text{P\&R},30}$ to denote the resulting max flows and the corresponding max-flow values are 13, 18, and 9 respectively. Let $A_{\text{P\&R},28}$, $A_{\text{P\&R},29}$, and $A_{\text{P\&R},30}$ denote the incidence matrices of the three max flows respectively. The final output $G_{\text{P\&R},\{28,29,30\}}$, the union of the three max-flows, will have an incidence matrix

$$A_{\text{P\&R},\{28,29,30\}} = \max(A_{\text{P\&R},28}, A_{\text{P\&R},29}, A_{\text{P\&R},30}).$$

There are 163 edges in $G_{\text{P\&R},\{28,29,30\}}$ and the total cost is

$$\sum_{e\in G_{\text{P\&R},\{28,29,30\}}} c(e) \approx 27.3294.$$

For comparison, the generalized coded feedback algorithm now takes only 344 seconds to converge. There are 142 edges

in the output graph $G_{\mathrm{Grdy},\{28,29,30\}}$ and the total cost is

$$\sum_{e\in G_{\mathrm{Grdy},\{28,29,30\}}} c(e) \approx 19.8500.$$

The power saving of the generalized coded feedback algorithm with greedy AB-MULTI-RE is 27.4% and the convergence time is only 38.1% of that of the three-staged P&R algorithm. The power saving follows from that *the generalized coded feedback algorithm is able to consider all destinations simultaneously* while the P&R algorithm searches for max flows in a sequential, one by one fashion without holistic consideration.

Note that, our solution outputs a *locally* min-cost multicast code. We also compute a *globally optimal* min-cost multicast code using linear programming, which has a total cost 18.2036. The penalty of searching for locally min-cost multicast network code is $\approx 9\%$. Note that our solution of searching for a locally min-cost solution is empirically faster (and asymptotically no slower) than the P&R algorithm, which is much faster than the linear-programming-based distributed network algorithms, as the latter requires carefully coordinated queue-length updates and exchanges with small step sizes.

VII. CONCLUSION

A new coded feedback max (s, d)-flow algorithm is provided for directed acyclic networks, which trims the network coding traffic by network coding. The algorithm is asymptotically no slower than the non-coded-communication-based push-&-relabel (P&R) algorithms, admits straightforward distributed network implementations, imposes no additional requirement on intermediate nodes, and sustains the optimal transmission rate even before the algorithm converges. The proposed coded feedback algorithm can also search for the minimal union of multiple max flows for the multicast scenario within the same time of finding a single max flow, a strict refinement over our preliminary work. The coded feedback scheme serves as a precursor to a new algorithmic study of network coding. We conclude this paper by a non-comprehensive list of future directions that we are actively investigating.

1) The coded feedback scheme provides a countermeasure to harness the power of broadcast. We are interested in the dynamics between this pair of opposite mechanisms. In a multiple unicast setup when several users are competing for network resources, a new rate-control scheme will be devised accordingly. We will also exploit this pair of mechanisms in non-stationary, time-varying networks. Jointly, the new rate-control algorithm and the results on time-varying networks will provide a new cross-layer framework of scheduling, route-finding, and coded feedback for wireless networks.
2) Improved pipelining: Waiting for *all* incoming messages to stabilize before starting trimming the graph is a bit conservative. In simulations, many graph-pruning opportunity can be identified even before all messages are stabilized. The properly designed pipelining will accelerate the convergence, which mitigates further the negative impact of initial broadcast.
3) For a practical network with delay, any directed cyclic graph is decoupled naturally into its acyclic counterpart along the time axis [11] and the coded feedback algorithm can be applied directly in a distributed fashion. Therefore, for network coding applications with generation-based structure, there is no advantage of considering cyclic networks. On the other hand, it is, in theory, an interesting and open question whether this network-coding-based approach can be generalized to cyclic networks as well, especially considering that the P&R algorithm is applicable to both acyclic and cyclic networks. To solve this question, new methods of constructing network coding for cyclic networks have to be investigated. The finding will complement our understanding of designing network-coding-based max flow algorithms.

REFERENCES

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*. Kyoto, Japan, August 2007.

[2] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, October 2003.

[3] C. Fragouli and A. Markopoulou, "A network coding approach to network monitoring," in *Proc. 43rd Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, USA, October 2005.

[4] C. Fragouli, A. Markopoulou, and S. Diggavi, "Topology inference using network coding," in *Proc. 44th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, USA, October 2006.

[5] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 829–848, March 2006.

[6] M. Gjoka, C. Fragouli, P. Sattri, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. 2007 Global Telecommunications Conference (GLOBECOM)*, November 2007, pp. 381–386.

[7] A. Goldberg, "Recent developments in maximum flow algorithms," NEC Research Institute, Inc., Technical Report 98-045, 1998.

[8] A. Goldberg and R. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM*, vol. 35, no. 4, pp. 921–940, October 1988.

[9] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.

[10] D. Karger and C. Stein, "A new approach to the minimum cut problem," *Journal of the ACM*, vol. 43, no. 4, pp. 601–640, July 1996.

[11] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, February 2003.

[12] C.-C. Wang, "Pruning network coding traffic by network coding — a new max-flow algorithm," in *Proc. IEEE Int'l Symp. Inform. Theory*. Toronto, Canada, July 2008.

[13] Y. Wu, M. Chiang, and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A critical cut approach," in *Proc. 2nd Workshop on Network Coding, Theory, & Applications (NetCod)*. Boston, Massachusetts, April 2006.

[14] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A shortest path approach," *IEEE J. Selected Areas of Commun.*, vol. 24, pp. 1475–1488, August 2006.