Optimal Scheduling with Network Coding for Relay-Aided Wireless Broadcast

Linyu Huang
Dept. of Electronic Engineering
City University of Hong Kong
linyhuang2@student.cityu.edu.hk

Chi Wan Sung
Dept. of Electronic Engineering
City University of Hong Kong
albert.sung@cityu.edu.hk

***Abstract*—This paper considers the problems of minimizing the completion time and reducing decoding complexity for relay-aided wireless broadcast. Both network coding and scheduling problems are considered. A deterministic network coding algorithm is designed to select innovative encoding vectors, which is applicable to both base station and relay. Compared with random linear network coding, the proposed algorithm can reduce decoding complexity significantly by selecting sparse encoding vectors. Integrating with the proposed network coding algorithm, a scheduling scheme based on dynamic programming is proposed, which is proved to be optimal in terms of minimizing expected completion time. Simulation shows that the proposed network coding algorithm and scheduling scheme work very well on both reducing completion time and decoding complexity.**

***Keywords*—wireless broadcast; network coding; scheduling**

I. Introduction

Linear network coding techniques have been widely applied to improve bandwidth efficiency and reliability of a network [1]. The source node sends the encoded packets together with the corresponding encoding vectors to all the receivers. An encoding vector is called *innovative to a receiver* if it is not in the subspace spanned by the previously received encoding vectors of that user. An encoding vector is said to be *innovative* if it is innovative to all the receivers that have not received enough packets for decoding. To minimize the *completion time*, i.e., the total time to complete the broadcast, the source node needs to find encoding vectors which are innovative to receivers as many as possible.

For relay-aided broadcast, some previous works consider instantly decodable network coding (IDNC) over $GF(2)$ [2], [3]. In [2], the authors propose a scheme based on IDNC. In [3], the authors propose another scheme, which is more efficient. However, innovative encoding vectors may not exist in the binary field, resulting in large number of retransmissions and long completion time. There are also some works, which consider random linear network coding (RLNC). In [4], the authors address the scheduling problem to minimize the completion time for a relay-aided broadcast system based on RLNC. They propose a greedy one-step scheduling method which requires instantaneous feedback after each retransmission, and a multi-step scheduling method based on dynamic programming which only requires feedback after multiple retransmissions. They also extend the methods to a two-cell broadcast system in [5]. For RLNC, although encoded packets broadcast by base station are almost always innovative when the finite field size is sufficiently large [6], [7], large field size brings high bandwidth overheads to carry the encoding vectors and increases the decoding computation complexity. Hence, it is preferable to work over finite fields with appropriate size. In [8]–[10], the authors propose different methods to find innovative encoding vectors for the base station, in which the field size is only required to be no less than the number of receivers. Unfortunately, since the relay may have only an incomplete set of (possibly coded) packets, those methods cannot be directly applied to the relay.

Another issue needs concern is how to schedule the transmission of the base station and the relay. In [2]–[5], the authors make a common assumption that the channel gains of relay-destination channels are always higher than that of source-destination channels, which makes the scheduling problems relatively simple. However, this assumption does not hold in many wireless broadcast applications.

In this paper, we consider minimizing completion time for relay-aided wireless broadcast with linear network coding. We derive a criterion to determine whether a relay has innovative encoding vectors or not over a finite field with size $q \geq K$, where K is the number of receivers. We extend the method of [9], so that it can be used to find an innovative encoding vector at a relay when $q \geq K$, provided that such a vector exists. In case when an innovative vector does not exist, our proposed method can find a vector that is simultaneously innovative to all those receivers to which the relay can provide new information. It also has a similar sparsity feature of the method proposed in [9], which can reduce the decoding complexity greatly when compared with RLNC. Based on our network coding method, we design a scheduling scheme based on dynamic programming. This joint scheduling and network coding scheme is proved to be optimal in the sense of minimizing the expected completion time.

II. System Model

As shown in Figure 1, we consider a time slotted wireless broadcast system consisting of a base station BS, a relay R

This work was partially supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

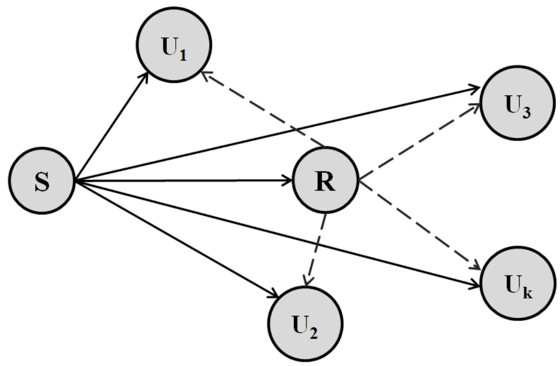

Figure 1. System model

and K users which are randomly located and labeled as U_i, where $i \in \{1, 2, \ldots, K\}$.

All downlink channels are modeled as mutually independent packet erasure channels. The downlink channels include the channels from BS to R, from BS to users and from R to users, which are characterized by erasure probabilities P_{sr}, P_{su_i} and P_{ru_i}, respectively, where $i \in \{1, 2, \ldots, K\}$. If a packet is erased, it will be discarded. Otherwise, the packet is assumed to be successfully received. The erasure probabilities are determined by the distances between corresponding senders and receivers. The uplink channels are assumed to be acknowledgement channels without error and delay. In other words, both BS and R know the status of R and all users.

We consider linear network coding over finite field $GF(q)$, where $q \geq (K+1)$. The broadcast file is divided into N independent equal-size packets, which are called *uncoded packets*. Each packet is a vector whose components are drawn from $GF(q)$. Each packet transmitted by BS is a linear combination over $GF(q)$ of the N uncoded packets. Each packet transmitted by R is a linear combination over $GF(q)$ of the packets it has received. In other words, each transmitted packet is a linear combination of the N uncoded packets, and the N coefficients are represented by a $1 \times N$ vector, which is called the *encoding vector* of that packet. When a coded packet is broadcast, its encoding vector will be attached in the header and broadcast at the same time. For clarity, we say broadcasting an encoding vector $\mathbf{x}$ to mean broadcasting a coded packet with encoding vector $\mathbf{x}$. Both BS and R can broadcast packets to users, but they are not allowed to transmit simultaneously to avoid collisions. For user U_i, we use r_i to denote the number of linearly independent encoding vectors it has received. By putting these r_i vectors together we obtain an $r_i \times N$ matrix $\mathbf{C}_i$, which we call it the *encoding matrix* of user U_i. Similarly, we use r_R and $\mathbf{C}_R$ to denote the number of linearly independent encoding vectors and encoding matrix of R, respectively. BS's encoding matrix $\mathbf{C}_{BS}$ can be regarded as the identity matrix $\mathbf{I}_N$. If the rank of U_i's/R's encoding matrix is N, it can decode the original file, and we say that U_i/R is *complete*. Otherwise, we say that U_i/R is *incomplete*.

For a set of parameters (such as N, K, P_{su_i}, P_{ru_i}, etc.), we use an integer random variable T to define the required number of transmissions to complete the broadcast. We define a channel realization of broadcast as the erasure conditions of all the channels under a particular realization.

Our first objective is to minimize the expected completion time $E[T]$ with linear network coding over $GF(q)$, where $q \geq (K+1)$, by determining the sender and transmitted encoding vector for each time slot. The expected completion time $E[T]$ is averaged over all the channel realizations. Our second objective is to reduce the decoding complexity by determining the encoding vectors transmitted by relay. The decoding complexity is defined as the average number of finite-field additions and multiplications that it takes to decode the broadcast packets for each user.

Throughout this paper, we use $RowSpace(\mathbf{C})$ to denote the row space of matrix $\mathbf{C}$. The null space of $\mathbf{C}$ is the orthogonal complement of $RowSpace(\mathbf{C})$. We use $Rank(\mathbf{C})$ to denote the rank of matrix $\mathbf{C}$. We use $[\mathbf{C}_1; \mathbf{C}_2]$ to denote the new matrix obtained by stacking $\mathbf{C}_2$ to the end of $\mathbf{C}_1$, i.e. $[\mathbf{C}_1; \mathbf{C}_2] = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}$.

III. Finding Encoding Vectors

When the sender has all the uncoded packets, e.g. BS, it can always find an encoding vector which is innovative to all the incomplete receivers when $q \geq K$. In [9], [10], the authors prove that there always exists a K-Sparse innovative vector, where K-Sparse means that the Hamming weight, i.e. the number of nonzero elements, of the $1 \times N$ innovative vector is less than or equal to K. They also provide a method to find such vectors. Their method is summarized as Algorithm 1, which is called Greedy Hitting (GH) method. However, their method is not directly applicable to the incomplete relay. We extend their method so that it is applicable to both BS and relay, no matter R is complete or not. Before that, we will first introduce the following useful theorem.

Theorem 1: Given some N-column matrices $\mathbf{C}$ and $\mathbf{C}_i$ over $GF(q)$, where $i \in \{1, 2, \ldots, K\}$. If $q \geq K$ and $Rank([\mathbf{C}_i; \mathbf{C}]) > Rank(\mathbf{C}_i)$ for all $\mathbf{C}_i$'s, there exists an innovative encoding vector which is in $RowSpace(\mathbf{C})$.

The proof of this theorem can be found in [11]. The following example illustrates the idea:

Example 1: . Let $q = 3, K = 2$ and $N = 4$. Consider

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 2 & 0 & 0 \end{bmatrix}, \mathbf{C}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \text{ and}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

By appending two linearly independent vectors which are also linearly independent to the row vectors of $\mathbf{C}$

$$[0\ 0\ 1\ 0] \text{ and } [0\ 0\ 0\ 1]$$

to the bottom of $\mathbf{C}$, we can get

$$\mathbf{C}_{coord} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

whose row vectors is an ordered new basis for $GF(q)^N$.

Algorithm 1: Greedy Hitting (GH) method (see [10])

Input: Given $\mathbf{C}_i$ over $GF(q)^N$ with $Rank(\mathbf{C}_i) < N$ and $q > K$, where $i \in \{1, 2, \ldots, K\}$.
Output: An K-sparse innovative vector $\mathbf{x}^*$ over $GF(q)^N$
Step 1: For each $\mathbf{C}_i$, find a basis for its null space and put those basis vectors together as row vectors to construct a matrix $\mathbf{B}_i$.
Step 2: For each $\mathbf{B}_i$, define a $1 \times N$ vector $\tilde{\mathbf{b}}_i$. The j-th element of $\tilde{\mathbf{b}}_i$ is set to 1 if the j-th column of $\mathbf{B}_i$ is nonzero, otherwise set to 0.
Step 3: Find the corresponding Hitting Set Problem and use greedy algorithm of [13] to solve it. The solution indicates the minimum number of nonzero elements of $\mathbf{x}^*$, and one possible combination of nonzero elements' positions. Denote the collection of those positions as a set P.
Step 4: For each $\mathbf{B}_i$, randomly select an row vector $\mathbf{b}'_i$ which has at least one nonzero elements in the positions listed in P.
Step 5: Construct an inequation $\mathbf{b}'_i \cdot \mathbf{x} \neq 0$ for each $\mathbf{b}'_i$. Solve the system of K inequations, and the solution is the required innovative vector $\mathbf{x}^*$.

We use the new coordinate system defined by the row vectors of $\mathbf{C}_{coord}$ to represent $\mathbf{C}$, $\mathbf{C}_1$ and $\mathbf{C}_2$, we can find

$$\mathbf{C}' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{C}'_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix} \text{ and}$$

$$\mathbf{C}'_2 = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where $\mathbf{C}' = \mathbf{C}\mathbf{C}_{coord}^{-1}$ and $\mathbf{C}'_i = \mathbf{C}_i\mathbf{C}_{coord}^{-1}$, $i = 1, 2$.

Then, for each $\mathbf{C}'_i$, i.e. $\mathbf{C}'_1$ and $\mathbf{C}'_2$, find a basis for its null space and put these basis vectors together as row vectors to construct a matrix $\mathbf{B}_i$. We can get

$$\mathbf{B}_1 = [0\ 1\ 0\ 0] \text{ and } \mathbf{B}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Choose the first rows of $\mathbf{B}_1$ and $\mathbf{B}_2$ and denote them as $\mathbf{b}_1$ and $\mathbf{b}_2$, respectively. We have

$$\mathbf{b}_1 = [0\ 1\ 0\ 0] \text{ and } \mathbf{b}_2 = [1\ 0\ 0\ 0].$$

In linear algebra, a vector $\mathbf{x}$ is in $RowSpace(\mathbf{C}'_i)$ if and only if $\mathbf{x} \cdot \mathbf{v} = 0$ for all vectors $\mathbf{v} \in RowSpace(\mathbf{B}_i)$. If we can find a vector $\mathbf{x} = [x_1\ x_2\ 0\ 0]$ such that

$$\mathbf{x} \cdot \mathbf{b}_1 \neq 0, \text{ and } \mathbf{x} \cdot \mathbf{b}_2 \neq 0,$$

$\mathbf{x}$ would be innovative to both U_1 and U_2. Actually, since both $\mathbf{b}_1$ and $\mathbf{b}_2$ cannot be a zero vector, such $\mathbf{x}$ always exists when $q \geq K$ (see [9, Lemma 3]).

Here, one possible solution is $\mathbf{x} = [1\ 1\ 0\ 0]$. The physical meaning of $\mathbf{x}$ is to add the first and second packets received by R to get the encoded packet. The coordinate of $\mathbf{x}$ relative to the original coordinate system is $\mathbf{x}^* = \mathbf{x}\mathbf{C}_{coord} = \mathbf{r}_1 + \mathbf{r}_2 = [1\ 0\ 0\ 1]$, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first and second rows of $\mathbf{C}_{coord}$, as well as that of $\mathbf{C}$. The vector $\mathbf{x}^* = [1\ 0\ 0\ 1]$ is the innovative vector we need. It means the encoded packet is equal to the sum of the first and fourth uncoded packets.

Algorithm 2: Greedy Hitting with Coordinate Change

Input: Given the sender's encoding matrix $\mathbf{C}$ and the receivers' encoding matrices $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K$ over $GF(q)^N$. $Rank([\mathbf{C}_i; \mathbf{C}]) > Rank(\mathbf{C}_i)$ for all $i \in \{1, 2, ..., K\}$.
Output: An innovative encoding vector $\mathbf{x}^*$ in $RowSpace(\mathbf{C})$
Step 1: Transform $\mathbf{C}$ to its RREF, denoted as $\mathbf{C}_{rref}$.
Step 2: If $\mathbf{C}_{rref} = \mathbf{I}_N$, use Algorithm 1 to find $\mathbf{x}^*$ and algorithm halts. Otherwise, continue.
Step 3: Let $r_c = Rank(\mathbf{C}_{rref})$. Extend $\mathbf{C}_{rref}$ to an ordered basis with the first r_c basis vectors being the row vectors of $\mathbf{C}_{rref}$. Put these ordered basis vectors together as row vectors to construct a matrix $\mathbf{C}_{coord}$.
Step 4: Represent $\mathbf{C}_{rref}$ and each $\mathbf{C}_i$ in the new coordinate system spanned by row vectors of $\mathbf{C}_{coord}$, denote as $\mathbf{C}' = \mathbf{C}_{rref}\mathbf{C}_{coord}^{-1}$ and $\mathbf{C}'_i = \mathbf{C}_i\mathbf{C}_{coord}^{-1}$, respectively.
Step 5: For each $\mathbf{C}'_i$, find a basis for its null space and put those basis vectors together as row vectors to construct a matrix $\mathbf{B}_i$.
Step 6: Only consider the first r_c columns of each $\mathbf{B}_i$, use Steps 2 $\sim$ 5 of Algorithm 1 to find a $1 \times r_c$ solution $\mathbf{x}$.
Step 7: Append $(N - r_c)$ zero elements to the end of $\mathbf{x}$ to get a $1 \times N$ vector $\mathbf{x}'$.
Step 8: Represent $\mathbf{x}'$ back into the coordinate system spanned by the standard basis to get the required innovative vector $\mathbf{x}^* = \mathbf{x}'\mathbf{C}_{coord}$.

Hence, R can broadcast the encoded packet with encoding vector $\mathbf{x}^*$, which is innovative to both U_1 and U_2.

Since the relay cannot find innovative vectors for U_i if $Rank([\mathbf{C}_i; \mathbf{C}_R]) \leq Rank(\mathbf{C}_i)$, the maximum number of users that a vector from $RowSpace(\mathbf{C}_R)$ can be simultaneously innovative to is the total number of those users with $Rank([\mathbf{C}_i; \mathbf{C}_R]) > Rank(\mathbf{C}_i)$. By Theorem 1, if R is incomplete, there always exists such an encoding vector that is innovative to all those users with $Rank([\mathbf{C}_i; \mathbf{C}_R]) > Rank(\mathbf{C}_i)$ if $q \geq K$. We generalize GH method so that it can find such encoding vectors at R. Formally, the extended algorithm can be described as Algorithm 2. We call it Greedy Hitting with Coordinate Change (GHCC).

IV. Three-Phase Scheduling Scheme

The broadcast is divided into three phases as follows.

Phase I: BS sequentially broadcasts each uncoded packet once to R and users.

Phase II: BS is selected to transmit innovative encoding vectors until the following condition is satisfied:

$$Rank([\mathbf{C}_i; \mathbf{C}_R]) = N \text{ for all } i \in \{1, 2, \ldots, K\}. \quad (1)$$

We call this condition the *Full Rank* condition. The encoding vectors are required to be not only innovative to R and all the users, but also out of $RowSpace([\mathbf{C}_i; \mathbf{C}_R])$ for all those users with $Rank([\mathbf{C}_i; \mathbf{C}_R]) < N$.

The encoding vectors in this phase are found as follows. First, construct a temporary encoding matrix $\mathbf{D}_i$ for each user U_i. For those users with $Rank([\mathbf{C}_i; \mathbf{C}_R]) < N$, let $\mathbf{D}_i$ be $[\mathbf{C}_i; \mathbf{C}_R]$. For other users, let $\mathbf{D}_i$ be $\mathbf{C}_i$. An encoding vector can then be found by applying GHCC with these N temporary encoding matrices $\mathbf{D}_i$ as input.

Phase III: Repeatedly select BS or R to broadcast until all users are complete. GHCC can be directly applied to BS and R in this phase. When finding encoding vectors at R, it only takes the users as its receiver. When finding encoding vectors at BS, besides the users, it also takes R as its receiver. Based on the Full Rank condition and Theorem 1, the found encoding vectors are always innovative to all the incomplete receivers.

To schedule the transmissions of BS and R in Phase III, we use dynamic programming to solve the problem recursively. We first define the following parameters.

- **State** We use a $1 \times K$ state vector $\mathbf{s} = [r_1\ r_2 \cdots r_K]$ to denote the status of all the users. The entire finite state space $\mathcal{S}$ can be denoted as $\mathcal{S} = [s_1\ s_2\ \ldots\ s_K]$, where $s_i \in \{r_i, r_i+1, \ldots, N\}, i \in \{1, 2, \ldots K\}$ and r_i is the rank of $\mathbf{C}_i$ when the broadcast enters Phase III. The goal state is $\mathbf{s}_g = [s_1\ s_2\ \ldots\ s_K]$, where $s_i = N$ for all $i \in \{1, 2, \ldots, K\}$.
- **Action** At each time step, we define the selected action a as $a \in \mathcal{A} = \{0, 1\}$. $a = 0$ denotes that BS is selected to transmit, and $a = 1$ denotes that R is selected.
- **State Transition Probability** At state s, the state transition probability that transfers from $\mathbf{s}$ to $\mathbf{s}'$ under action a is denoted as $p(\mathbf{s}'|\mathbf{s}, a)$. Since the channels are assumed to be independent, $p(\mathbf{s}'|\mathbf{s}, a)$ can be easily computed. Due to space limitation, the details are skipped here.

Given a state $\mathbf{s}$, we define $V(\mathbf{s})$ as the expected number of transmissions for the system to evolve from $\mathbf{s}$ to $\mathbf{s}_g$ given that optimal actions are chosen in every state, and define $\pi(\mathbf{s})$ as the optimal action at state $\mathbf{s}$. It is clear that $V(\mathbf{s})$ is non-zero except when $\mathbf{s} = \mathbf{s}_g$. Given a state $\mathbf{s} \in \mathcal{S} \setminus \{\mathbf{s}_g\}$, we define $V(\mathbf{s}, a)$ as the expected number of transmissions for the system to evolve from $\mathbf{s}$ to $\mathbf{s}_g$ given that action a is chosen for state $\mathbf{s}$ and optimal actions are chosen for the other states. $V(\mathbf{s}, a)$ can be computed by solving the following equation:

$$V(\mathbf{s}, a) = 1 + \sum_{\mathbf{s}' \in \mathcal{S} \setminus \{\mathbf{s}\}} p(\mathbf{s}'|\mathbf{s}, a) V(\mathbf{s}') + p(\mathbf{s}|\mathbf{s}, a) V(\mathbf{s}, a), \quad (2)$$

where the number 1 means each state transition takes one transmission. Note that the summation only involves those states that can be immediately reached from $\mathbf{s}$. In other words, state $\mathbf{s}'$ is involved only when $p(\mathbf{s}'|\mathbf{s}, a)$ is non-zero. Since $p(\mathbf{s}|\mathbf{s}, a)$ is a constant, equation (2) can be written as:

$$V(\mathbf{s}, a) = \Big[1 + \sum_{\mathbf{s}' \in \mathcal{S} \setminus \{\mathbf{s}\}} p(\mathbf{s}'|\mathbf{s}, a) V(\mathbf{s}')\Big] / [1 - p(\mathbf{s}|\mathbf{s}, a)]. \quad (3)$$

Note that our problem has the property that the state-transition graph is a directed acyclic graph. It means that given two distinct states, $\mathbf{s}$ and $\mathbf{s}'$, if $\mathbf{s}'$ can be reached from $\mathbf{s}$ after a certain number of state transitions, then $\mathbf{s}$ cannot be reached from $\mathbf{s}'$. For directed acyclic graph, it is well known that there is a topological ordering of vertices, which means that if there is a state transition from $\mathbf{s}$ to $\mathbf{s}'$, then $\mathbf{s}$ comes before $\mathbf{s}'$ in the ordering. Due to this property, $V(\mathbf{s})$ can be computed backward from $V(\mathbf{s}_g)$. To compute $V(\mathbf{s})$, we apply the formula in (3) to find $V(\mathbf{s}, a)$ for $a \in \mathcal{A} = \{0, 1\}$. $V(\mathbf{s})$ and the optimal action at $\mathbf{s}$ can then be obtained by

$$V(\mathbf{s}) = \min_{a \in \mathcal{A}} V(\mathbf{s}, a) \text{ and } \pi(\mathbf{s}) = \arg\min_{a \in \mathcal{A}} V(\mathbf{s}, a),$$

respectively. To implement this idea, recursion may be used, and we state the recursive algorithm in Algorithm 3. The proposed GHCC with the three-phase scheduling based on dynamic programming is called GHCC-DpS, and has the following property.

Algorithm 3: Scheduling by Dynamic Programming

Input: A state $\mathbf{s} \in \mathcal{S} \setminus \{\mathbf{s}_g\}$
Output: $V(\mathbf{s})$ and $\pi(\mathbf{s})$ for all $\mathbf{s} \in \mathcal{S}$

set $V(\mathbf{s}) := 0$ and $\pi(\mathbf{s}) := 0$ for $\mathbf{s} \in \mathcal{S}$ //global variables
do OptimalAction(s)
return $V(\mathbf{s})$ and $\pi(\mathbf{s})$ for all $\mathbf{s} \in \mathcal{S}$

Function: OptimalAction(s)
  for $\mathbf{s}' \in \mathcal{S} \setminus \{\mathbf{s}_g, \mathbf{s}\}$ **do**
    if $p(\mathbf{s}'|\mathbf{s}, a) > 0$ for any $a \in \mathcal{A}$ **and** $V(\mathbf{s}') = 0$
      do OptimalAction(s′)
    end if
  end for
  for $a \in \mathcal{A}$ **do**
    $V(\mathbf{s}, a) := [1 + \sum_{\mathbf{s}' \in \mathcal{S} \setminus \{\mathbf{s}\}} p(\mathbf{s}'|\mathbf{s}, a) V(\mathbf{s}')] / [1 - p(\mathbf{s}|\mathbf{s}, a)]$
  end for
  $V(\mathbf{s}) := \min_{a \in \mathcal{A}} V(\mathbf{s}, a)$
  $\pi(\mathbf{s}) := \arg\min_{a \in \mathcal{A}} V(\mathbf{s}, a)$
end function

Theorem 2: GHCC-DpS is an optimal strategy for minimizing the expected completion time, $E[T]$.

The proof of this theorem is lengthy and is omitted here. We refer the readers to [11] for a detailed proof.

V. Numerical Results

In this section, we evaluate the performance of our proposed scheme via simulations. A 90-degree sector of a single cell with a radius of 10 Km is considered. BS is located at the center of the cell. Both R and users are randomly located within the sector area. The channels are modeled containing combined effects of path loss and Rayleigh fading. The background noise P_N is assumed to be -113 dBm. The erasure probabilities are defined as the probabilities that the received signal-to-noise ratio (SNR) is below some threshold S_{th}. The instantaneous SNR γ is distributed according to an exponential distribution with parameter $\frac{1}{\bar{\gamma}}$ [12], where $\bar{\gamma}$ is the average SNR. Here, the average SNR can be written as $\bar{\gamma} = \frac{GP_T}{d^3 P_N}$, where P_T is the transmit power, d is the distance between the corresponding transmitter and receiver, and G is the gain resulted from antenna configurations and other losses. Hence, the erasure probabilities can be computed as

$$P = P(\gamma < S_{th}) = 1 - e^{-d^3 P_N/(GP_T)S_{th}}.$$

The parameters are set as follows. P_T is set to be 47 dBm for both BS and R, G equals 3 for the channel from BS to R and 1 for all other channels. S_{th} is set to 10 dB.

We compare the proposed GHCC and scheduling schemes with the traditional automatic repeat request (ARQ), IDNC [3] and RLNC [4].To be fair, we consider ARQ with the following optimal scheduling. BS first sequentially broadcasts each uncoded packet once to users and R. Then in the retransmission phase, each uncoded packet will be repeatedly transmitted until all the users are complete. For a packet that will be transmitted in the retransmission phase, it will be broadcast by BS if R does not have it. Otherwise, it will be transmitted by the one with $\min\{\max\{P_{su_i}\}, \max\{P_{ru_i}\}\}$, where the two maximum operations are taken over the set $\{i : U_i$ does not have the uncoded packet to be transmitted$\}$. For the IDNC scheme in [3], BS first sequentially broadcasts each uncoded packet once. In the second phase, BS transmits coded packets until R has enough information to serve all users. Then it enters phase three, in which R transmits coded packets until the broadcast is complete. For RLNC, since the one step scheduling of [4] always outweighs the multi-step scheduling of [4], we only compare with the former one, which is denoted as RLNC with the One Step Scheduling (RLNC-OSS). RLNC-OSS first selects BS to transmit N randomly *coded* packets in Phase I. Then, it defines two parameters: $B_{BS} = \sum_{i=1}^{K} \lambda_{su_i} P_{su_i}$ and $B_R = \sum_{i=1}^{K} \lambda_{ru_i} P_{ru_i}$ for BS and R, respectively. At a particular time, $\lambda_{su_i}(\lambda_{ru_i})$ is set to one if the rank of BS(R)'s encoding matrix is greater than that of U_i, otherwise set to zero. At each slot of the retransmission phase, it selects the one with $\max\{B_{BS}, B_R\}$ to transmit a coded packet, until all users are complete.

The performance of a joint network coding and scheduling method is measured in terms of completion time and decoding complexity. The completion time is defined as the total number of transmissions. As most of the previous works, we use Gauss-Jordan elimination for decoding. The decoding complexity is compared in term of total number of additions and multiplications that used by all users. When measuring the decoding complexity, an addition operation is counted if none of the operands is nonzero, and a multiplication is counted if none of its operands is 1 or 0. The computations of a user include that used to check whether a received encoding vector is innovative to itself and that used to decode the packets. Note that IDNC is operated in the binary field, no multiplication is required. For each set of parameters, we take the average of 10,000 random realizations as the final result.

We ran simulations with $N = 3$ and K varies from 2 to 7 over $GF(11)$. Simulation results are shown in Figures 2–4. Figure 2 is the comparison of completion time, which shows that GHCC-DpS has shorter completion time than other schemes. Compared with ARQ, IDNC and RLNC-OSS, GHCC-DpS can reduce completion time by about 14%, 6% and 4%, respectively, when $K = 7$. The comparisons of decoding complexity are shown in Figure 3 and 4. Since ARQ does not require decoding, it is not plotted. The figures show that GHCC-DpS requires significantly fewer multiplications and additions than RLNC-OSS. Although GHCC-DpS requires serval multiplications in each broadcast, it requires much less additions than IDNC.

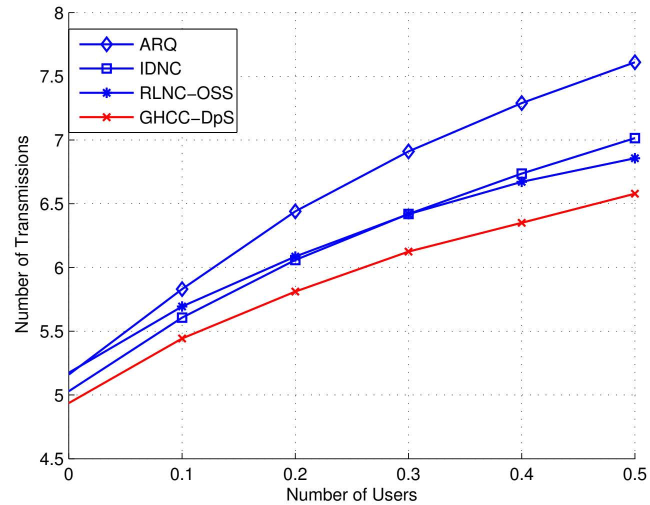

Figure 2. Completion time for various K with $q = 11, N = 3, P_{sr} = 0.25, P_{su_i} = 0.5$ and $P_{ru_i} = 0.1$ for all users

We also run simulations with the same parameters corresponding to Figure 2 of [4], where $N = 6, K = 2, q = 3, P_{su_i} = 0.5$, $P_{ru_i} = 0.1$ for all users, and P_{sr} varies from 0 to 0.5. The simulation results are shown in Figure 5. Note that RLNC-OSS-2 is under the assumptions of [4] that the field size is sufficiently large, and an encoding vector is assumed to be innovative to a user if the rank of sender's encoding matrix is greater than that of the user's. It can be regarded as a lower bound for RLNC-OSS with larger field size. We can see that GHCC-DpS still has shorter completion time than other schemes. In particular, compared with RLNC-OSS, ARQ, RLNC-OSS-2 and IDNC, GHCC-DpS can reduce completion time by about 18%, 10%, 8%, and 2%, respectively, when $P_{sr} = 0.5$. The comparisons of decoding complexity are similar to Figure 3 and 4, and are omitted here.

VI. Conclusion

In this paper, we first derive a criterion to determine whether a (possibly incomplete) relay has innovative vectors when $q \geq K$. If innovative vectors exist, our proposed method, GHCC, can determine one of them. Otherwise, it can find a vector that is innovative to all receivers to which the relay is able to provide new information. These findings are fundamentals of many data dissemination applications.

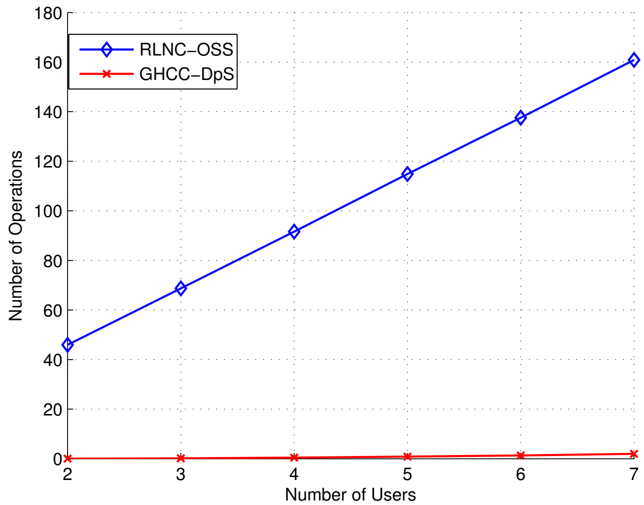

Figure 3. Multiplication operations for different K with $q = 11, N = 3, P_{sr} = 0.25, P_{su_i} = 0.5$ and $P_{ru_i} = 0.1$ for all users

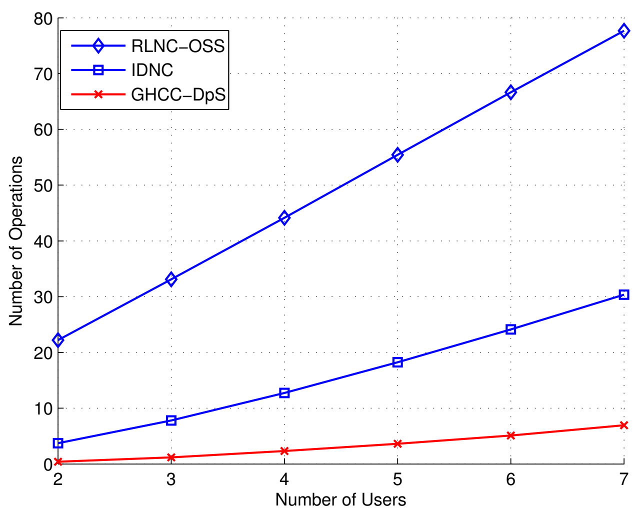

Figure 4. Addition operations for different K with $q = 11, N = 3, P_{sr} = 0.25, P_{su_i} = 0.5$ and $P_{ru_i} = 0.1$ for all users

We apply our network coding results to the design of a relay-aided wireless broadcast system, which issues a new scheduling problem when compared with the design of a one-to-many broadcast system. Based on our design of the network code and the statistical properties of the transmission channels, we derive an optimal scheduling algorithm using dynamic programming. This algorithm, called GHCC-DpS, is able to minimize the expected completion time. This result provides fundamental understanding of optimal operation of a network-coded relay-aided system. Simulation results shown that GHCC-DpS outweigh the traditional RLNC in terms of both completion time and decoding complexity, and outweigh IDNC in terms of completion time.

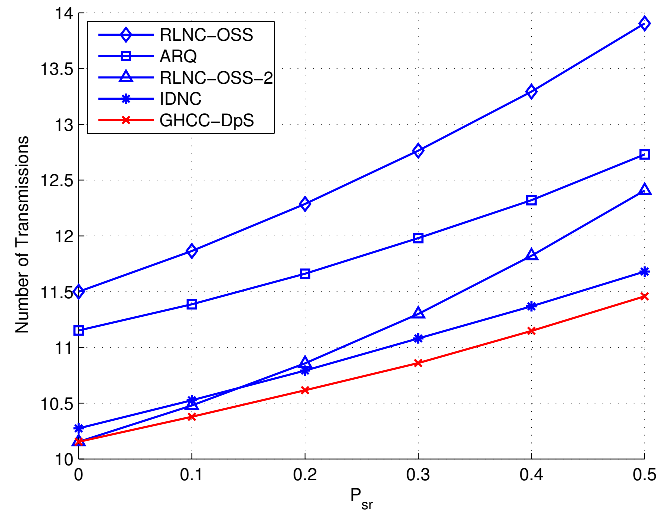

Figure 5. Completion time for various P_{sr} with $K = 2, q = 3, N = 6, P_{su_i} = 0.5$ and $P_{ru_i} = 0.1$ for all users

ACKNOWLEDGMENT

The authors would like to thank Dr. Kenneth Wing-Ki Shum for helpful discussions, and Dr. Ho Yuet Kwan for helpful discussions and providing useful program modules.

REFERENCES

[1] S. Y. R. Li, R. W. Yeung and N. Cai, "Linear network coding," in *IEEE Trans. Inf. Theory*, pp. 371-381, Jun. 2003.

[2] P. Fan, C, Zhi, C, Wei and K. B. Letaief, "Reliable relay assisted wireless multicast using network coding," in *IEEE J. Sel. Areas Commun.*, pp. 749-762, Jun. 2009.

[3] L. Lu, M. Xiao and K. L. Rasmussen, "Design and analysis of relay-aided broadcast using binary network codes," in *J. of Commun.*, vol. 6, no. 8, pp. 610-617, Nov. 2011.

[4] L. Lu, M. Xiao, K. L. Rasmussen, and M. Skoglund, "Efficient scheduling for relay-aided broadcasting with random network codes," in *Proc. IEEE PIMRC*, Toronto, Canada, Sep. 2011, pp.1825-1829.

[5] L. Lu, F. Sun, M. Xiao and L. K. Rasmussen, "Relay-aided multi-cell broadcasting with random network coding," in *Int. Symp. on Inf. Theory and its Applicat(ISITA)*, pp. 957-962, Dec. 2010.

[6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," in *IEEE Trans. Inf. Theory*, pp. 4413-4430, 2006.

[7] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT*, pp. 442-442, 2003.

[8] H. Y. Kwan, K. W. Shum, and C. W. Sung, "Generation of innovative and sparse encoding vectors for broadcast systems with feedback," to appear in *Proc. IEEE ISIT*, Saint Petersburg, Russia, Aug. 2011.

[9] C. W. Sung, K. W. Shum, and H. Y. Kwan, "On the sparsity of a linear network code for broadcast systems with feedback," in *2011 Int. Symp. on NetCod*, Beijing, China, Jul. 2011, pp. 1-4.

[10] H. Y. Kwan, K. W. Shum, and C. W. Sung, "Linear network code for erasure broadcast channel with feedback: complexity and algorithms," arXiv:1205.5324v1, May 2012.

[11] L. Huang and C. W. Sung, "Scheduling and network coding for relay-aided wireless broadcast: optimality and heuristic", preprint.

[12] M. K. Simon and M. S. Alouini, "Fading channel characterization and modeling", in *Digital communication over fading channels a unified approach to performance analysis*, 2nd ed., Wiley, 2005.

[13] U. Feige, "A threshold of ln n for approximating set cover", in *J. of ACM 45*, pp. 643-652, 1998.