Probabilistic Models and Algorithms for Data Synchronization/Broadcast via Network Coding

Duong Nguyen-Huu, Thinh Nguyen, *Member, IEEE*

***Abstract*—We investigate the problem of data synchronization in which a sender has a set of packets to be distributed to all the receivers via a broadcast channel. Initially, each receiver has some fraction of the packets. At each time slot, the sender might broadcast a packet to all the receivers. The goal is to find a broadcast scheme that minimizes the number of time slots until all the receivers successfully obtain all the packets. We propose two probabilistic models on how the initial fractions of packets at receivers are distributed. These models arise naturally in many large scale systems such as Peer-to-Peer (P2P) networks, data centers, and distributed storage systems. Based on these models, we establish probabilistic bounds and asymptotic results on the minimum number of time slots to successfully transmit all the packets to all the receivers. Next, we propose and analyze a number of random network coding algorithms for finding the approximately optimal solution. Theoretical analysis and simulations are provided to verify the probabilistic bounds and the proposed algorithms.**

***Index Terms*—Network Coding, Galois Field, Data Synchronization, Index Coding Problem, Probability.**

I. Introduction

Data synchronization plays a critical role in the performances of many emerging large scale distributed systems such as Peer-to-Peer (P2P) systems, distributed storage systems, and data centers. To provide high reliability in such systems, data are typically duplicated across multiple nodes in a network. In addition, many systems allow data to be updated asynchronously at individual nodes. As a result, potential data inconsistencies might arise across multiple nodes. For example, during the peak time, a data center [1], [2], [3] might allow data to be updated at individual servers autonomously for better performance. These changes are then propagated to other servers at an appropriate later time. During this interval, the data across the servers are inconsistent. In other systems, data inconsistencies at different nodes are resulted in a far less controllable way. Notably, in file sharing systems such as BitTorrent, peers might have different parts of the same file due to the random exchange of data among peers. Wireless broadcast is another example in which many users receive the same file broadcast from a base station. However, due to packet losses, for some given time, users might have different parts of the file. Thus, the aim of the data synchronization problem is to repair the data inconsistencies by broadcasting additional data to the receivers.

The authors are with the School of Electrical Engineering and Computer Science, Oregon State University, Oregon, OR, 97331 USA (e-mail:nguyendu, thinhq@eecs.oregonstate.edu).

The data synchronization problem is an instance of the index coding problem [4], [5], [6] that consists of a sender and a number of receivers sharing a common broadcast channel. The sender has a set of packets $\mathcal{A}$. Each receiver has a random subset of $\mathcal{A}$. At each time slot, the sender broadcast a packet that can be received by all the receivers. The goal is to find a broadcast scheme that minimizes the number of time slots until every receiver successfully receive the set $\mathcal{A}$. An approach to this problem is to use the Network Coding (NC) framework. NC framework treats each packet as an element in a finite field. Each coded NC packet is a linear combination of other packets. It is shown that when the finite field size is larger than or equal to the number of nodes, the problem can be solved in polynomial time [7], [8]. However, for arbitrary field size, the synchronization problem appears to be similar to the multicast network coding problem which has been shown to be NP-hard when the field size is smaller than the number of receivers [9], [10].

Contributions. In this paper, we study the synchronization problem from a probabilistic viewpoint. First, we describe two probabilistic models on how subsets of packets at receivers are distributed. These models arise naturally in many large scale systems such as Peer-to-Peer (P2P) networks, data centers, and distributed storage systems. For these two models, we establish probabilistic bounds and asymptotic results on the minimum number of time slots that the sender needs to successfully transmit all the packets to all receivers. Such bounds can shed lights on the benefits and limitations of using NC-based broadcast schemes in certain real-world settings. Second, while the probabilistic upper and lower bounds for the optimal solution can be found, finding the algorithms for achieving the optimal solution is not trivial. Therefore, we propose and analyze a number of random network coding (RNC) algorithms for finding the optimal solutions. Our analysis provides quantitative performances in terms of expectation, variance, and tail probability on the number of time slots required to complete the synchronization for the proposed algorithms.

Outline. We first discuss a few related work in Section II, then present the problem formulation and notations in Section III. In Section IV, we describe two common models in which data inconsistency can occur. Based on these models, we show the probabilistic bounds on the optimal solutions of any broadcast scheme in Section V. We then describe three NC-based algorithms to perform synchronization and their theoretical performance analysis in Section VI and Section VII. In Section VIII, we provide the simulation results for the proposed algorithms and finally a few concluding remarks in

Section IX.

II. Related Work

There exists rich literature on NC on which our work is built upon. Due to limited space, we will discuss the similarities and differences between our work and a few representative work. Our work is closely related to the index coding problem [4], which has been shown to be NP-hard [11], [12], [13], [14], [15] and a number of heuristic schemes have been proposed [16], [17]. Both problems consists of a number of receivers who want to receive an identical set of packets $\mathcal{A}$ from a sender. All the receivers share the same broadcast channel, and have different subsets of $\mathcal{A}$. The goal of both problems is to minimize the number of broadcasts by the sender until all the receivers successfully obtain the complete set $\mathcal{A}$. On the other hand, our work differs in the following ways. First, instead of assuming the subsets of packets at the receivers are given as in most network coding literature [4][18], we propose two probabilistic models to characterize the distribution of the subsets of packets. Based on these two models, we further study the asymptotic bounds on the optimal solution which, to our knowledge, has not been investigated previously. Specifically, we study how the number of packets varies as a function of the number of receivers as both become large, can affect the solution. Second, instead of solving the problem in a deterministic manner, we propose randomized NC algorithms to find the approximate optimal solution with probabilistic guarantees.

We note that in many existing NC literature, the information about the partial sets of packet at the receiver is assumed to be available at server statically. For many large scale distributed systems consisting many users and large data, this assumption might be impractical since the central server might need to store a substantial large amount of information. Instead, we introduce three algorithms that allow different levels of information exchange between the sender and the receivers dynamically. That said, our work is on the simplicity of randomized network coding techniques [19], [18], [20], [21] that can be implemented in real world settings. In addition, our theoretical results have probabilistic flavor as contrast to the work in [22].

Our work can also be viewed as an instance of the Direct Data Exchange (DDE) problem that was first proposed by El Rouayheb et al. [23]. The DDE problem has attracted much interest from the research community [24] [25] [26] [27]. While the goal of both problems is to synchronize data in multiple receivers, there are essential differences. In the the DDE problem, all the receivers have to participate in broadcasting their data while in our problem, only one sender can broadcast. In addition, in the DDE problem, the subset of packets at each receiver can only be original packets while in our setup, we allow both mixed (network coded) and original packets in the subsets of the packets. Furthermore, most existing solutions to the DDE problem take a deterministic approach while ours has a probabilistic flavor.

That said, our work is very similar to the problem of wireless broadcast using network coding via lossy channel. For example, in a single-hop wireless network, where communication channels are lossy, NC techniques are used to help the receivers to recover the lost packets quickly [28], [29]. In wireless ad hoc network, NC techniques have been also applied to increase bandwidth efficiency [30], [31]. In wireless mesh network, the advantages of NC compared to traditional approach are presented [32], [33]. Majority of these schemes use the XOR operation since it can be implemented efficiently in practice. Our work extends the analysis and performance characterization of NC using a general finite field. It is motivated by the well-developed theory of linear network code [34], [35] and the robustness of applying random linear network coding into multicast application [36], [37].

III. Problem Formulation

A. Problem Description and Notation

Consider the following broadcast scenario with one sender who wants to broadcast two packets $\mathbf{p}_1$ and $\mathbf{p}_2$ to two receivers R_1 and R_2. We assume R_1 has packet $\mathbf{p}_1$ while R_2 has packet $\mathbf{p}_2$. The goal is to minimize the number of broadcasts by the sender so that each receiver will have both packets $\mathbf{p}_1$ and $\mathbf{p}_2$, hence their data is synchronized. A straightforward way is for the sender to broadcast $\mathbf{p}_1$ first then $\mathbf{p}_2$. Assuming no packet loss, R_1 and R_2 will have both packets in two time slots. However, a better way is for the sender to broadcast only one packet $\mathbf{c} = \mathbf{p}_1 \oplus \mathbf{p}_2$ where $\oplus$ denotes bit-wise exclusive OR of bits in the two packets. Upon receiving $\mathbf{c}$, R_1 and R_2 will be able to recover their missing packets, respectively as: $\mathbf{c} \oplus \mathbf{p}_1 = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \mathbf{p}_1 = \mathbf{p}_2$, and $\mathbf{c} \oplus \mathbf{p}_2 = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \mathbf{p}_2 = \mathbf{p}_1$. This example illustrates the benefit of network coding, i.e., mixing packets appropriately to reduce the number of broadcasts. In general, the problem of finding a broadcast scheme, i.e., the right "coded" packets that minimizes the number of transmissions for an arbitrary number of users with an arbitrary pattern of packets is an NP-hard problem [22]. As such, we consider a probabilistic approach to this problem as described shortly. We now use the following notations to describe the problem.

- There is one sender with a set of D original packets denoted as $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_D\}$, and N receivers denoted as $R_1, R_2, \ldots R_N$ that want to obtain these D packets.
- Each receiver R_i has a "Has" set $\mathcal{H}_i$ consisting of exactly $K_i \leq D$ packets. Denote $\mathcal{W}_i = \mathcal{P} \setminus \mathcal{H}_i$ as the "Want" set of packets that the receiver R_i wants but does not have.
- A network coded (mixed) packet $\mathbf{c}$ is constructed as:

$$\mathbf{c} = v_1\mathbf{p}_1 + v_2\mathbf{p}_2 + \cdots + v_D\mathbf{p}_D \tag{1}$$

  with $v_i \in GF(\mathcal{F})$. Each $\mathbf{p}_i$ can be viewed as an element in $GF(\mathcal{F}^D)$. Consequently, we can view a packet as a row vector $\mathbf{v} = (v_1, v_2, \ldots, v_D)$, and the "Has" set $\mathcal{H}_i$ as a matrix $\mathbf{H}_i$ whose rows are $\mathbf{v}$'s. Also, for brevity, we denote $F = |\mathcal{F}|$.
- At each time slot, the sender is allowed to broadcast exactly one mixed or original packet to all the receivers. Furthermore, we assume no packet loss during broadcast.

- Let T_i denote the number of time slots until the receiver R_i receives a sufficient number of packets to be able to reconstruct all D original packets.
- Let $T = \max\{T_1, T_2, \dots T_N\}$ denote the number of time slots until all the receivers are able to decode all the D original packets.

Note that a receiver will be able to reconstruct all the D original packets if it collects any D packets (mixed or original) that span a D dimensional space. Specifically, recall that a packet can be represented as a row vector $\mathbf{v}_i$, then if the matrix

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_D \end{pmatrix}$$

has rank D (full rank), then the original packets $\mathbf{p}_i$'s can be reconstructed via solving a set of linearly independent equations.

For simplicity, the packet length as defined above is artificially constrained to length $\lceil D \log F \rceil$ bits. In practice, a packet should be a vector of length $n \gg D$ whose each element is $\lceil D \log F \rceil$ bits long. Thus the number of bits to specify v_i in the packet header (necessary for the receivers to decode) is negligible. Finally, we note that the optimal scheme is the one that minimizes T.

B. Example

We now give an example to illustrate the notations and concepts. Let $D = 4, K_1 = K_2 = 2, \mathcal{F} = \{0, 1\}$, and thus $GF(2)$ is used for all the finite field computations. A receiver R_1 has $\mathcal{H}_1 = \{\mathbf{p}_1 \oplus \mathbf{p}_3, \mathbf{p}_2\}$, and its initial "Has" set $\mathcal{H}_1$ can be represented as a matrix:

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Note that since $GF(2)$ is used, each entry in the matrix can only be 0 or 1. If the sender broadcasts two packets $(\mathbf{p}_2 \oplus \mathbf{p}_4)$ and $\mathbf{p}_3$ which collectively can be represented as a matrix $\mathbf{S}$ below.

$$\mathbf{S} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Assuming no packet loss, then the new "Has" set $\hat{\mathcal{H}}_1$ of receiver R_1 would have two more elements. Thus the corresponding new matrix $\hat{\mathbf{H}}_1$ is:

$$\hat{\mathbf{H}}_1 = \begin{pmatrix} \mathbf{H_1} \\ \mathbf{S} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Since the $rank(\mathbf{H}_1) = 4$ (full rank) in $GF(2^4)$, R_1 can reconstruct all original packets $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$.

As described, an optimal broadcast scheme is one with the minimum number of transmissions that enables all the receivers to obtain their corresponding full rank matrices. Clearly, the minimum number of transmissions depends on the initial "Has" sets of each receivers. We note that a typical setting of network coding problem assumes that the sender has complete information about the "Has" sets of each receivers initially. On the other hand, in this paper, we will describe algorithms that allow different levels of information exchange between the sender and the receivers. In the next section, we will describe two models of the "Has" sets that arise naturally from real-world settings. We then use these models to characterize the optimality of the solutions for any broadcast scheme via probabilistic bounds in Section V.

IV. Models of the "Has" Set

We consider two models for the "Has" sets at individual receivers. We call these the "uncoded" and "coded" models of the "Has" set. These models aim to approximate the real-world scenarios.

Uncoded Model. The first model can be used to approximate a wireless broadcast scenarios in WiFi or cellular networks. Specifically, in the "Uncoded" model, each individual receivers has K_i original packets out of the D original packets $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_D$, where K_i is random variable drawn from the Binomial distribution with parameters (D, α_i). This model arises from considering a scenario in which a sender broadcast D original packets to N receivers. Due to different channel conditions, each receiver has a different probability α_i of receiving the packets. Assuming that the outcomes of the D transmissions are independent across packets and receivers, then the number of packets received at the receivers follow D independent Binomial distributions. Starting at this point, the sender can employ an optimal transmission scheme that ensure all N receivers can receive all the D original packets in minimum number of transmissions. The optimal solution depends on the pattern of packets at the receivers, i.e., their "Has" sets. While finding the optimal scheme is NP-hard, given the probability model of the "Has" sets, it is possible to characterize the optimal solution, i.e., the minimum number of transmissions via probabilistic bounds as will be shown in Section V. These bounds are useful in the sense that one can bound the optimal solution without knowing the optimal transmission scheme. Furthermore, in some cases, it is possible to determine whether network coding scheme is even useful.

Coded Model. In the "coded" model, each receiver is to assume to have S packets. However, these packets are network coded packets, defined previously as:

$$\mathbf{c} = v_1\mathbf{p}_1 + v_2\mathbf{p}_2 + \cdots + v_D\mathbf{p}_D.$$

Each coded packet is drawn randomly at uniform from $F^D - 1$ possible coded packets independently without replacement. The "Coded" model can be used to represent data stored Peer-to-Peer (P2P) network. In this setting, a file is first broken into D packets, then a number of coded packets are produced using coefficients v_i drawn uniformly at random. These coded packets are then distributed to the peers via some P2P transmission protocols. Each peer can also mix the packets it receives and forwards the mixed (coded) packets to another peers. As a result, the S packets stored at a peer can be thought as S coded packets drawn randomly at uniform from the $F^D - 1$ possible coded packets.

V. Probabilistic Characterization of Optimal Solution

In this section, we characterize the minimum number of packets to be sent by the sender in order for all the receivers to recover all the packets, regardless of the algorithms used to find the right packets to send. We first discuss the trivial bound on the minimum number of transmissions T^* needed for the N receivers to recover all D original packets with each receiver R_i having its "Has" set $\mathcal{H}_i$. We then derive the probability distribution of T^* when packets in the set $\mathcal{H}_i$ are drawn according to the "Uncoded" and "Coded" models described in the previous section. In some cases, it is sufficient and simple to use the probability bounds, rather than a full distribution to characterize the optimality. We will provide these probabilistic bounds as well.

Supposed there are N receivers $R_1, R_2, \dots R_N$, each has a number of packets, i.e., "Has" set $\mathcal{H}_i$ which can be represented as a matrix $\mathbf{H}_i$. Let K_i be the rank of $\mathbf{H}_i$, and let $K = \min\{K_i\}$. Then it is easy to see that the minimum number of transmissions T^* needed for the N receivers to recover all D original packets must be upper and lower bounded by

$$D - K \leq T^* \leq D \tag{2}$$

The trivial bound, however does not take the advantage of probabilistic models, thus can be quite loose. Next, we characterize the full probability distributions of $T_l^* = D - K$ and give probabilistic bounds on T_l^*. Notably, we use these bounds to determine the effectiveness of any network coding scheme in the "Uncoded" model.

A. Analysis of the "Uncoded" Model

We will first determine the distribution of K, then the distribution of T_l^* can be completely characterized. However the closed-form distribution is a bit complicated that prevents us from drawing a good intuition. Therefore, we also provide probabilistic bounds for K that allows us to draw a better intuition.

1) Computing Distribution of K: To derive the distribution, we note that K_i is a Binomial random variable with D being the number of trials and α_i the probability of success. Thus, we have:

$$\begin{cases} Rank(\mathbf{H}_i) = K_i \\ \mathbf{P}(K_i = k) = f(k, D, \alpha_i) = C_D^k \alpha_i^k (1-\alpha_i)^{D-k} \\ \mathbf{P}(K_i \leq k) = F(k, D, \alpha_i) = \sum_{j=0}^{k} C_D^j \alpha_i^j (1-\alpha_i)^{D-j} \end{cases} \tag{3}$$

Now since $K = \min\{K_i\}$, one can find the cumulative probability distribution of K as follows.

$$F_K(k) = \mathbf{P}(K \leq k) = 1 - \prod_i^N (1 - \mathbf{P}(K_i \leq k)) \tag{4}$$

Then the probability distribution of K can be computed from the cumulative function:

$$\begin{aligned} \mathbf{P}(K = k) &= F_K(k) - F_K(k-1) && (5) \\ &= \prod_{i=1}^{N} (1 - P(K_i \leq k-1)) \\ &\quad - \prod_{i=1}^{N} (1 - P(K_i \leq k)) && (6) \\ &= \prod_{i=1}^{N} (1 - \sum_{j=0}^{k-1} C_D^j \alpha_i^j (1-\alpha_i)^{D-j}) \\ &\quad - \prod_{i=1}^{N} (1 - \sum_{j=0}^{k} C_D^j \alpha_i^j (1-\alpha_i)^{D-j}) && (7) \end{aligned}$$

We can see that the closed-form distribution does not provide a good intuition. Hence, we now provide some probabilistic bounds regarding K.

2) Probabilistic Bounds for K: Let $\alpha_{min} = \min\{\alpha_i\}$ and $\alpha_{max} = \max\{\alpha_i\}$. We have following Proposition regarding the tail bound for K.

Proposition 1. *(Tail bound) For $0 < k < D\alpha_{min}$, we have*

$$\boldsymbol{P}(K > k) \geq (1 - \exp(-\frac{1}{2\alpha_{min}} \frac{(D\alpha_{min} - k)^2}{D}))^N \tag{8}$$

Proof. We have:

$$\begin{aligned} \mathbf{P}(K > k) &= \prod_{i=1}^{N} \mathbf{P}(K_i > k) && (9) \\ &= \prod_{i=1}^{N} (1 - \mathbf{P}(K_i \leq k)) && (10) \\ &= \prod_{i=1}^{N} (1 - F(k, D, \alpha_i)) && (11) \end{aligned}$$

Also, $F(k, D, \alpha_{min}) \geq F(k, D, \alpha_i)$. Hence,

$$\mathbf{P}(K > k) = (1 - F(k, D, \alpha_i))^N \geq (1 - F(k, D, \alpha_{min}))^N \tag{12}$$

Also by Chernoff's inequality, we have:

$$F(k, D, \alpha_{min}) \leq \exp(-\frac{1}{2\alpha_{min}} \frac{(D\alpha_{min} - k)^2}{D})$$

Plug in (12), we complete the proof. □

Since $T_l^* = D - K$, Proposition 1 indicates that minimum number of retransmission for the "Uncoded" model depends on the receiver with the smallest probability of successful packet reception.

Next, we have the following proposition regarding the asymptotic behavior of D and N.

Proposition 2. *(Asymptotic) For $N \to \infty$ and any k, α_{min} such that $0 < k < D\alpha_{min}$, we have:*

$$\begin{cases} \boldsymbol{P}(K > k) \to 0 \text{ for } D = o(\log(N)) \\ \boldsymbol{P}(K > k) \to c \text{ where } c \in (0,1) \text{ for } D = \Theta(\log(N)) \\ \boldsymbol{P}(K > k) \to 1 \text{ for } D = \omega(\log(N)) \end{cases} \tag{13}$$

(Using Bachmann-Landau notations for $o(), \Theta(), \omega()$).

Proof. We first show the case when $D = \Theta(\log(N))$.

$$\begin{aligned}\exp\left(-\frac{1}{2\alpha_{min}}\frac{(D\alpha_{min}-k)^2}{D}\right) &= exp(-\Theta(\log(N)) \\ &= \Theta(\frac{1}{N}) \leq c_1\frac{1}{N}\end{aligned}$$

for some $0 < c_1 < \infty$. Hence,

$$(1-\exp\left(-\frac{1}{2\alpha_{min}}\frac{(D\alpha_{min}-k)^2}{D}\right))^N \geq (1-\frac{c_1}{N})^N. \quad (14)$$

Now,

$$\lim_{N\to\infty}(1-\frac{c_1}{N})^N = e^{-c_1}, \quad (15)$$

and from (8), (14), and (15), when $N \to \infty$, we obtain

$$\mathbf{P}(K > k) \geq e^{-c_1} > 0. \quad (16)$$

We note that in (16), $\mathbf{P}(K > k)$ is strictly greater than 0.

On the other hand, using $\binom{D}{l} \geq 1$, we have

$$\begin{aligned}F(k, D, \alpha_{max}) &= \sum_{l=0}^{k}\binom{D}{l}\alpha_{max}^l(1-\alpha_{max})^{D-l} \quad (17)\\ &\geq \sum_{l=0}^{k}\alpha_{max}^l(1-\alpha_{max})^{D-l} \quad (18)\\ &= (1-\alpha_{max})^D\sum_{l=0}^{k}(\frac{\alpha_{max}}{1-\alpha_{max}})^l \quad (19)\\ &\geq (1-\alpha_{max})^D. \quad (20)\end{aligned}$$

Since $D = \Theta(\log(N))$, and $0 < 1-\alpha_{max} < 1$, we have

$$(1-\alpha_{max})^D = (1-\alpha_{max})^{\Theta(\log(N))} = \Theta(\frac{1}{N}).$$

Hence, $F(k, D, \alpha_{max}) \geq c_2(\frac{1}{N})$ for some $\infty > c_2 > 0$. Therefore,

$$(1-F(k, D, \alpha_{max}))^N \leq (1-\frac{c_2}{N})^N = e^{-c_2} < 1$$

for $N \to \infty$. Similar to (12), we have:

$$\mathbf{P}(K > k) = (1-F(k, D, \alpha_i))^N \leq (1-F(k, D, \alpha_{max}))^N \quad (21)$$

Combine these two above equations, we have:

$$\mathbf{P}(K > k) < 1. \quad (22)$$

Now, from (16) and (22), we have $0 < \mathbf{P}(K > k) < 1$. This completes the proof for $D = \Theta(\log(N))$.

For the case $D = \omega(\log(N))$, similarly we have:

$$\begin{aligned}\exp\left(-\frac{1}{2\alpha_{min}}\frac{(D\alpha_{min}-k)^2}{D}\right) &= \exp(-\omega(\log(N)) \\ &= \omega(\frac{1}{N}) \leq c_3\frac{1}{N}.\end{aligned}$$

for any $\infty > c_3 > 0$. Now,

$$\begin{aligned}\mathbf{P}(K > k) &\geq (1-\exp\left(-\frac{1}{2\alpha_{min}}\frac{(D\alpha_{min}-k)^2}{D}\right))^N \\ &\geq (1-\frac{c_3}{N})^N \to e^{-c_3} \to 1 \quad (23)\end{aligned}$$

for $N \to \infty$ and $c_3 \to 0$.

Also $\mathbf{P}(K > k) \leq 1$, then $\mathbf{P}(K > k) \to 1$ for $D = \omega(\log(N))$.

Finally, for $D = o(\log(N))$, similarly we have

$$\begin{aligned}F(k, D, \alpha_{max}) &\geq (1-\alpha_{max})^D = (1-\alpha_{max})^{o(\log(N))} \\ &= o(\frac{1}{N}) \geq c_4\frac{1}{N}\end{aligned}$$

for any $\infty > c_4 > 0$. Hence,

$$\mathbf{P}(K > k) \leq (1-F(k, D, \alpha_{max}))^N \leq (1-\frac{c_4}{N})^N \to e^{-c_4} \to 0 \quad (24)$$

for $N \to \infty$ and $c_4 \to \infty$.

Also $\mathbf{P}(K > k) \geq 0$ then $\mathbf{P}(K > k) \to 0$ for $D = o(\log(N))$. □

Using the parameters $\alpha_{min} = 0.3; \alpha_{max} = 0.7; k = \frac{D\alpha_{min}}{2}$, Fig. 1 shows the empirical probability $\mathbf{P}(K > k)$ that is accurately predicted by Proposition 2.

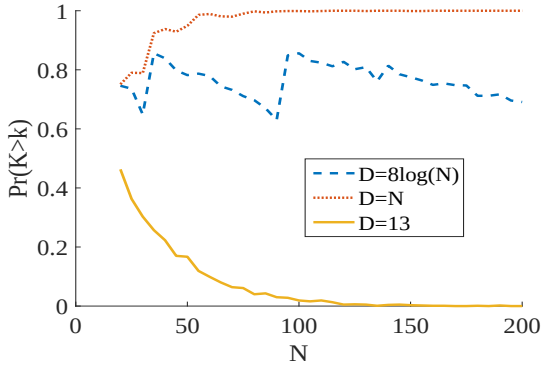

Figure 1: Empirical $\mathbf{P}[K > k]$ vs. N

There is an interesting point implied by Proposition 2. If the number of packets sent (D) is on the order of log of the number of receivers (N), then the probability $\mathbf{P}(K > k)$ does not approach 0 or 1 when N and D approach infinity. Rather, this probability approaches a number between 0 and 1. On other the hand, probability $\mathbf{P}(K > k)$ approaches 0 or 1 when the $D = o(\log(N))$ and $D = \omega(\log(N))$, respectively. Essentially, this implies that there is a phase transition that depends on on how large D is, compared with N in the asymptotic sense.

Consider the special case where $k = 0$, we have:

$$\begin{aligned}\mathbf{P}(K = 0) &= 1 - \mathbf{P}(K > 0) \\ &= 1 - \prod_{i=1}^{N}(1-F(0, D, \alpha_i) \\ &\geq 1 - (1-(1-\alpha_{max})^D)^N.\end{aligned}$$

From the above equation, we have the following corollary:

Corollary 1. *For fixed the number of packets D, and the number of receivers $N \geq \log_{[1-(1-\alpha_{max})^D]}\epsilon$,*

$$\mathbf{P}(K = 0) \geq 1 - \epsilon$$

where $\epsilon > 0$.

The corollary above can be interpreted as follows. When the number of receivers is sufficiently large, there exists a receiver which hasn't received any packet with almost certainty. Therefore, the senders needs to re-send all packets. Also in this scenario, the lower bound equals the upper bound $T_l^* = D - K = D = T_u^*$ which implies that network coding does not bring any benefit.

B. Analysis of the "Coded" Model

In the "coded" model, each receiver stores S vectors and each vectors would be drawn randomly in $GF(F^D)$ (including both original and combined packets). First, we need to find the distribution of $K_i = rank(\mathbf{H}_i)$ and then one can compute the distribution of K by using order statistics. Still, the formula is too complex. Hence, we also provide upper bound for expectation of K_i. By these bounds, one can establish bound for K by Markov's inequality.

1) Computing Distribution of K: Consider any receiver R_i, we choose randomly S vectors in $GF(F^D)$ and there are $rank(\mathbf{H}_i) = K_i$ linearly independent vectors. We can compute $\mathbf{P}(K_i = k)$ by a recursive approach as follows.

Consider any node i, let $f(k, s)$ be the probability that $S = s$ and $rank(\mathbf{H}_i) = K_i = k$.

Obviously, we can have (for simple cases):

$$\begin{cases} f(0,0) = 1 \\ f(k,s) = \prod_{j=1}^{j=k} p_j \text{ for } 1 \le k = s \le S \\ f(k,s) = 0 \text{ for } k > s. \end{cases} \tag{25}$$

The probability that we have k linearly independent vectors after picking up s random vectors is equal to sum of two probability: first is the probability that we have $k-1$ linearly independent vectors in $s-1$ random vectors and the s-th vector is linearly independent with existing vectors; second is the probability that we have k linearly independent vectors in $s-1$ random vectors and the s-th vector is dependent with existing vectors.

We have the inductive step for $0 < k < s \le S$ as follows.

$$f(k,s) = f(k-1, s-1)p_k + f(k, s-1)(1 - p_{k+1}) \tag{26}$$

where

$$p_j = 1 - \frac{F^{j-1} - 1}{F^D - 1} = \frac{F^D - F^{j-1}}{F^D - 1} \tag{27}$$

We can rewrite the function $f(k, s)$ as follows. In case $s < k$, $f(k, s) = 0$. In case $s \ge k$, we have

$$\begin{aligned} f(k,s) &= \sum_{\sum_j^{k+1} \alpha_j = s-k} \left(\prod_{i=1}^{k} p_i \prod_{j=2}^{k+1} (1-p_j)^{\alpha_j}\right) && (28) \\ &= \prod_{i=1}^{k} p_i \sum_{\sum_j^{k+1} \alpha_j = s-k} \left(\prod_{j=2}^{k+1} (1-p_j)^{\alpha_j}\right) && (29) \end{aligned}$$

where $\alpha_j = 0, 1, \ldots, s-k$ for $2 \le j \le k+1$. From the above formula, one can apply order statistics for i.i.d discrete variables [38] to compute the distribution of K.

2) Probabilistic Bounds on K:

Proposition 3. *(Tail bound) For $S \le D$ and any $k > 0$, we have:*

$$\boldsymbol{P}(K_i \ge k) \le \frac{(\beta^S - 1)}{(\beta - 1)} \frac{1}{k} \tag{30}$$

where $\beta = \frac{F^D - F}{F^D - 1}$

Proof. Let denote $E_j = \mathbf{E}[K_i | S = j]$ be the expected rank of matrix $\mathbf{H}_i$ given that $\mathbf{H}_i$ has j rows (or the number of independent packets). Let q_j be the probability that the j-th row is linearly independent with the previous $j-1$ rows.

$$\begin{aligned} E_j &= q_j(E_{j-1} + 1) + (1 - q_j)E_{j-1} \\ &= q_j + E_{j-1} \\ &= q_j + q_{j-1} + E_{j-2} \\ &\cdots \\ &= \sum_{j=1}^{i} q_j \end{aligned}$$

since $E_0 = 0$.

Now, in each receiver R_i, we have S packets. Hence,

$$E_S = \sum_{j=1}^{S} q_j \tag{31}$$

Now, consider q_j. The necessary condition for j-th row to be linearly independent with previous $j-1$ rows is that j-th row needs to be linearly independent with each row in $j-1$ rows.

$$q_j \le \left(\frac{F^D - F}{F^D - 1}\right)^{j-1} \tag{32}$$

for $j \ge 1$.

Combine (31) and (32), we have:

$$\sum_{j=1}^{S} \left(\frac{F^D - F}{F^D - 1}\right)^{j-1} \ge E_S = \mathbf{E}[K_i] \tag{33}$$

Hence, we can have an uppper bound U for $\mathbf{E}[K_i]$:

$$U = \sum_{j=1}^{S} \left(\frac{F^D - F}{F^D - 1}\right)^{j-1} = \frac{\beta^S - 1}{\beta - 1}$$

where $\beta = \frac{F^D - F}{F^D - 1}$.

One now can use Markov's inequality to complete the proof. □

Using the parameters $S = 3; F = 2; k = 2$, Fig. 2 shows the empirical probability $\mathbf{P}(K_i > k)$ and the upper bound for different values of D that match the prediction of the Proposition 3.

Since $K = \min\{K_i\}$, we have: $P(K \ge k) \le P(K_i \ge k)$. However, one can establish a tighter bound for K by applying the inequality for N independent random variables with identical mean and variance in [39].

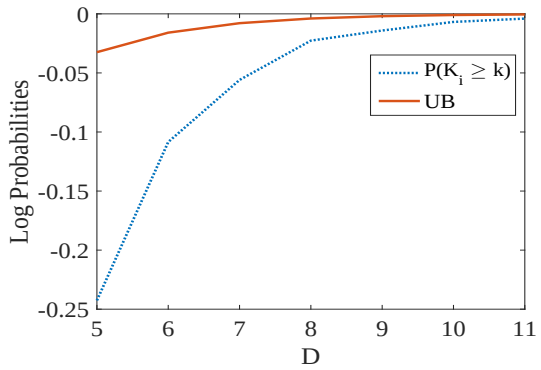

Figure 2: Empirical $\mathbf{P}[K > k]$ vs. D

Proposition 4. *(Asymptotic) Consider where $D \to \infty$ and $T = D - S$ is a constant, using the result given in [40, Theorem 1], we have:*

$$\lim_{D\to\infty} \boldsymbol{P}(K_i = k) = \begin{cases} \prod_{j=T+1}^{\infty} (1 - (\frac{1}{F})^j) \text{ for } k = 0 \\ \frac{\prod_{j=T+k+1}^{\infty} (1-(\frac{1}{F})^j)}{\prod_{j=1}^{k} (1-(\frac{1}{F})^j)} (\frac{1}{F})^{k(T+k)} \text{ for } k \geq 1 \end{cases}$$

Hence, one can compute the probability distribution of K by using order statistics.

VI. Algorithms

In the previous section, we characterize the optimal solution via asymptotic and probabilistic results. In this section, we describe three random network coding algorithms to approximate the optimal solution: the Simple Random Network Coding Algorithm (SRNC), the Informed Random Network Coding Algorithm (IRNC), and the Refined Random Network Coding Algorithm (RRNC). We note that the SRNC and IRNC algorithms are not novel. However, the probabilistic analysis for SRNC and IRNC with respect to the "Has" set models have not been done previously. We start with the simplest one: SRNC algorithm.

A. Simple Random Network Coding Algorithm (SRNC)

The SRNC algorithm is described as follows.

Algorithm 1: SRNC Algorithm

Data: The sender has no knowledge about packets at receivers
1 **while** *there exists one receiver that can't recover the original packets* **do**
2 Sender generates and broadcasts a mixed packet;
3 Each receiver R_i updates its "Has" set and corresponding matrix H_i;
4 **if** H_i *is full rank* **then**
5 R_i can recover the original packets and sends acknowledgment to the sender;
6 **end**
7 **end**

SRNC algorithm assumes that the sender has no knowledge about the subsets of packets at the receivers at any given time. At every time slot, the sender broadcasts a mixed packet (line 2)

$$\mathbf{c} = v_1\mathbf{p}_1 + v_2\mathbf{p}_2 + \cdots + v_D\mathbf{p}_D,$$

where v_i's are drawn uniformly at random from the finite field $GF(\mathcal{F})$. The sender will continue to broadcast these packets until it receives all the acknowledgments from each receiver, indicating that all the receivers have successfully obtained all the packets.

At the receiver, upon receiving a mixed packet $\mathbf{c}$, the "Has" set of a receiver R_i is updated as:

$$\mathcal{H}_i = \mathcal{H}_i \bigcup \{\mathbf{c}\},$$

and the corresponding matrix $\mathbf{H}_i$ is constructed (line 3). Next, the Gaussian elimination algorithm is applied to $\mathbf{H}_i$ to find linearly independent columns and the missing original packets. If $\mathbf{H}_i$ is full rank, then receiver R_i can recover the original packets. In this case, the receiver sends an acknowledgment to the sender indicating that it has successfully recovered all the original packets (line 5). Otherwise, it waits for the next packet from the sender. The process repeats until the receiver is able to recover all the original packets. The SRNC algorithm is simple since the sender does not require information from the receivers. Rather, only one acknowledgement from each receiver is sufficient to complete the synchronization process.

B. Informed Random Network Coding (IRNC)

The IRNC algorithm is described as follows.

Algorithm 2: IRNC Algorithm

Data: The sender has knowledge about "Want" sets at receivers only in the beginning
1 **while** *there exists one receiver that cannot recover the original packets* **do**
2 Sender generates and broadcasts a mixed packet based on the initial "Want" sets at receivers;
3 Each receiver R_i updates its "Has" set and corresponding matrix $\mathbf{H}_i$;
4 **if** H_i *is full rank* **then**
5 R_i can recover the original packets and sends an acknowledgment to the sender;
6 **end**
7 **end**

The IRNC algorithm requires a bit more information. Specifically, all receivers send the information on their "Want" sets to the sender only once in the beginning. The sender uses this information to construct and broadcast the mixed packets without further collaboration from the receivers except the final acknowledgements from each receiver indicating that they have successfully obtained all the packets.

The "Want" set at each receiver R_i is constructed as follows. First, the Gaussian elimination algorithm is applied to $\mathbf{H}_i$ to find the missing original packets. Next, R_i sends this information to the sender. The sender then constructs a union set $\mathcal{W} = \bigcup_i \mathcal{W}_i$ where $\mathcal{W}_i$ consisting of the missing original

packets for R_i. Let $M = |\mathcal{W}|$, the sender broadcasts a mixed packet constructed as:

$$\mathbf{c} = v_1\mathbf{p}_1 + v_2\mathbf{p}_2 + \cdots + v_M\mathbf{p}_M,$$

where $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M \in \mathcal{W}$, and v_i's are drawn uniformly at random from the finite field $GF(\mathcal{F})$. The only difference between IRNC and SRNC algorithms is that the IRNC algorithm generates mixed packets from $\mathcal{W}$ (line 2) while the SRNC algorithm generates mixed packets from all the original packets in $\mathcal{P}$.

As an example, consider a scenario with five original packets and two receivers R_1 and R_2. R_1 has three packets, each is a linear combination of the five original packets. Thus, $\mathbf{H}_1$ is a 3×5 matrix of the form:

$$\mathbf{H}_1 = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{pmatrix},$$

where * denotes values from $GF(\mathcal{F})$. Assume that $\mathcal{F} = \{0, 1\}$, then each row in $\mathbf{H}_1$ represents a packet of R_1 which is a linear combination of the five original packets. Since $GF(2^5)$ is used, a 1 or a 0 in the i-th column and j-th row indicates that the original packet $\mathbf{p}_i$ is present or not in the j-th mixed packet, respectively. Now R_1 applies the Gaussian elimination algorithm, and suppose it produces the following upper diagonal matrix:

$$\mathbf{H}_1' = \begin{pmatrix} 1 & * & * & * & * \\ 0 & 0 & 1 & * & * \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

Based on $\mathbf{H}_1'$, the "Want" set of R_1 includes $\mathbf{p}_2$ and $\mathbf{p}_4$. R_1 then sends this information to the sender. Similarly, if R_2's "Want" set contains only $\mathbf{p}_3$, it will send this information to the sender. The sender will now generate the mixed packets that are random linear combinations from the set $\mathcal{W} = \{\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$.

Receivers in the IRNC algorithm also behaves similarly to those in the SRNC algorithm. Since the IRNC algorithm generate packets based on the missing packets at the receivers, the sender avoids sending redundant information to the receivers. Therefore, the IRNC algorithm should perform better than the SRNC algorithm.

C. Refined Random Network Coding Algorithm (RRNC)

We now introduce the RRNC algorithm which can be shown theoretically better than the SRNC and IRNC algorithms. The RRNC algorithm is described as follows.

Compare to the previous two algorithms, the RRNC algorithm requires a bit more information exchange between the sender and receivers, but they all are very similar. Specifically, the sender receives the information on the "Want" sets from each receiver after transmitting each packet. It then constructs the union set $\mathcal{W} = \bigcup_i \mathcal{W}_i$, and generates mixed packets based on $\mathcal{W}$ in the exact manner as the IRNC algorithm. The only difference is that after receiving a new packet, each receiver recomputes its "Want" set and sends its updated "Want" set to the sender (line 7). The sender then constructs a new $\mathcal{W}$ and uses it to generate and broadcast the next packet (line 2). The process repeats until all the receivers can successfully recover all the original packets.

Algorithm 3: RRNC Algorithm

Data: The sender has knowledge about "Want" sets at receivers at each time slot

1 **while** *there exists one receiver that cannot recover all the original packets* **do**
2 Sender generates and broadcasts a mixed packet based on the "Want" sets;
3 Each receiver R_i updates its "Has" set and corresponding matrix $\mathbf{H}_i$;
4 **if** H_i *is full rank* **then**
5 R_i can recover the original packets and sends acknowledgment to the sender;
6 **else**
7 R_i computes and sends its "Want" set to the sender;
8 **end**
9 **end**

Intuitively, the RRNC algorithm is better than the IRNC and SRNC algorithms because at each time slot, the RRNC algorithm uses more information about the missing packets at each receiver. As a result, a mixed packet generated by the RRNC algorithm has a higher chance of adding more new information to the receivers than the others two. We will show the theoretical analysis in the next section.

VII. Theoretical Performance of the Proposed Algorithms

In this section, we provide a number of theoretical results on the performances for the proposed SRNC, IRNC, and RRNC algorithms in terms of the number of time slots for completing the data synchronization. First the performances of algorithms are considered from the viewpoint of a single receiver R_i. Recall in Section III that a packet can be represented as a vector $\mathbf{v}$. Thus, a group of packets are considered as mutually linearly independent if theirs vector representations are mutually linearly independent. We now consider a receiver R_i who wants to recover all $D = |\mathcal{P}|$ original packets. Given that R_i currently obtains $K \leq D$ linearly independent packets, we want to know on average how many time slots it takes for R_i to recover all D original packets using the SRNC, IRNC, and RRNC algorithms.

A. Single User's Perspective

Let $T_i^{(S)}$, $T_i^{(I)}$, and $T_i^{(R)}$ be the random variables denoting the number of packets sent out by the sender, i.e., the number of time slots required so that R_i is able to recover all the original D packets using the SRNC, IRNC, and RRNC algorithms, respectively. Let denote $|\mathcal{F}| = F$ and also $L = D - K$ be the cardinality of the individual "Want" set for each receiver R_i. Then, we have the following Propositions to characterize the performances of the proposed algorithms.

Proposition 5. *(Performance of the SRNC algorithm)*

$$\mathbf{E}[T_i^{(S)}] = \sum_{j=1}^{L} \frac{F^D - 1}{F^D - F^{K+j-1}} \tag{34}$$

$$\mathbf{Var}[T_i^{(S)}] = \sum_{j=1}^{L} \frac{(F^{K+j-1} - 1)(F^D - 1)}{(F^D - F^{K+j-1})^2} \tag{35}$$

Let $M = |\mathcal{W}|$ be the cardinality of the combined "Want" set, then the performance of the IRNC algorithm is characterized by the following Proposition.

Proposition 6. *(Performance of the IRNC algorithm)*

$$\boldsymbol{E}[T_i^{(I)}] = \sum_{j=1}^{L} \frac{F^M - 1}{F^M - F^{M-L+j-1}} \tag{36}$$

$$\boldsymbol{Var}[T_i^{(I)}] = \sum_{j=1}^{L} \frac{(F^{M-L+j-1} - 1)(F^M - 1)}{(F^M - F^{M-L+j-1})^2} \tag{37}$$

Next, the following Proposition characterizes the performance of the RRNC algorithm.

Proposition 7. *(Performance of the RRNC algorithm)*

$$\boldsymbol{E}[T_i^{(R)}] \leq \sum_{j=1}^{L} \frac{F^{M-j+1} - 1}{F^{M-j+1} - F^{M-L}} \tag{38}$$

$$\boldsymbol{Var}[T_i^{(C)}] \leq \sum_{j=1}^{L} \frac{(F^{M-j+1} - 1)(F^{M-L} - 1)}{(F^{M-j+1} - F^{M-L})^2}. \tag{39}$$

The proofs of all these Propositions can be found in the Appendix.

The following Proposition supports our intuition that the RRNC algorithm is better than the IRNC algorithm which in turn is better than the SRNC algorithm.

Proposition 8. *(Performance Comparison)*

$$\boldsymbol{E}[T_i^{(R)}] \leq \boldsymbol{E}[T_i^{(I)}] \leq \boldsymbol{E}[T_i^{(S)}] \tag{40}$$

$$\boldsymbol{Var}[T_i^{(R)}] \leq \boldsymbol{Var}[T_i^{(I)}] \leq \boldsymbol{Var}[T_i^{(S)}]. \tag{41}$$

Proof. For the expected value, let us consider the following function:

$$f(x) = \frac{F^x - 1}{F^x - F^{x-a}},$$

where $1 \leq a \leq L$ is a constant. We have:

$$f'(x) = \frac{lnF}{F^x - F^{x-a}} > 0.$$

where $x > a$. Therefore, $f(x)$ is a monotonically increasing function in x.

Now, from (34), (36), (38) the upper bound of $\mathbf{E}[T_i^{(R)}]$, $\mathbf{E}[T_i^{(I)}]$ and $\mathbf{E}[T_i^{(S)}]$ is the sum of functions of the form $f(M-j+1)$, $f(M)$ and $f(D)$, respectively. Also, $M-j+1 \leq M \leq D$. Thus, we have $\mathbf{E}[T_i^{(R)}] \leq \mathbf{E}[T_i^{(I)}] \leq \mathbf{E}[T_i^{(S)}]$.

For the variance, consider the following function:

$$g(x) = \frac{(F^x - 1)(F^{x-a} - 1)}{F^x - F^{x-a}} \tag{42}$$

where $1 \leq a \leq L$ is a constant. We have

$$g'(x) = \frac{\ln(F)(F^{2x} - F^a)}{F^x(F^a - 1)} > 0 \tag{43}$$

where $x > a$. Hence, $g(x)$ is a monotonically increasing function in x.

Now, from (35), (37), (39) the upper bound of $\mathbf{Var}[T_i^{(R)}]$, $\mathbf{Var}[T_i^{(I)}]$, and $\mathbf{Var}[T_i^{(S)}]$ is the sum of functions of the form $g(M-j+1)$, $g(M)$ and $g(D)$, respectively. Also, $M-j+1 \leq M \leq M$. Thus, we have $\mathbf{Var}[T_i^{(R)}] \leq \mathbf{Var}[T_i^{(I)}] \leq \mathbf{Var}[T_i^{(S)}]$. □

B. Sender's Perspective

We now consider the performance of the entire system, i.e., the sender's perspective. Let $T_{max}^{(S)}$, $T_{max}^{(I)}$, and $T_{max}^{(R)}$ be the random variables denoting the numbers of time slots until the sender receives all the acknowledgments from all N receivers using the SRNC, IRNC, and RRNC algorithms, respectively. Then clearly,

$$T_{max}^{(S)} = \max_i T_i^{(S)} \tag{44}$$

$$T_{max}^{(I)} = \max_i T_i^{(I)} \tag{45}$$

$$T_{max}^{(R)} = \max_i T_i^{(R)}, \tag{46}$$

for $i = 1, 2, \ldots, N$.

The performances of all three algorithms are characterized by the following Proposition.

Proposition 9. *(Tail probability)*

$$\mathbf{P}(T_{max} > a) \leq 1 - (1 - \frac{\sigma^2}{(a-\mu)^2})^N \tag{47}$$

for $a > \mu = \mathbf{E}[T_i]$ and $\sigma^2 = \mathbf{Var}[T_i]$ for each algorithm, respectively.

Alternatively, one can find the upper bound of $\mathbf{E}[T_{max}]$ by applying the inequality for N independent random variables with identical mean and variance in [39] as follows.

$$\mathbf{E}[T_{max}] \leq \mu + \sigma\sqrt{N-1}. \tag{48}$$

VIII. Performance Results

In this section, we present the performance evaluations of the proposed algorithms for various settings, and verify the agreement between the theoretical and empirical results.

Fig. 3 shows the empirical $\mathbf{E}[T_i^{(S)}]$, $\mathbf{E}[T_i^{(I)}]$, $\mathbf{E}[T_i^{(R)}]$, i.e., the average numbers of time slots needed for a receiver to recover all $N = 50$ original packets using the SRNC, IRNC, and RRNC algorithms, respectively, as a function of K, the number of packets initially at a receiver. The value range for K is from 20 to 30. As seen, the value of $\mathbf{E}[T_i^{(S)}]$ and $\mathbf{E}[T_i^{(I)}]$ are not much different to each other, while $\mathbf{E}[T_i^{(R)}]$ is slightly smaller. This complies with our intuition since the RRNC algorithm has more information than the others two. Despite of a modest improvement in mean of time slots needed to recover all the original packets for the RRNC algorithm, we note that the variance of $T_i^{(R)}$ is also smaller than those of the others

two. This is quite important as we consider the performance from the sender's perspective as shown in Fig. 4.

Fig. 4 shows empirical $\mathbf{E}[T_{max}^{(S)}]$, $\mathbf{E}[T_{max}^{(I)}]$, $\mathbf{E}[T_{max}^{(R)}]$ as the numbers of time slots needed for the sender to complete the synchronization process for the SRNC, IRNC, and RRNC algorithms. Now, one can see that the RRNC algorithm achieves a much better performance than those of the others two. We argue that this is due to smaller variance produced by the RRNC algorithm. This can be seen from Eq. (48) that $\mathbf{E}[T_{max}]$ for all three algorithms depends on the square root of the number of the receivers times the variance. Thus, a small change in variance can greatly affect $\mathbf{E}[T_{max}]$ for a large number of receivers.

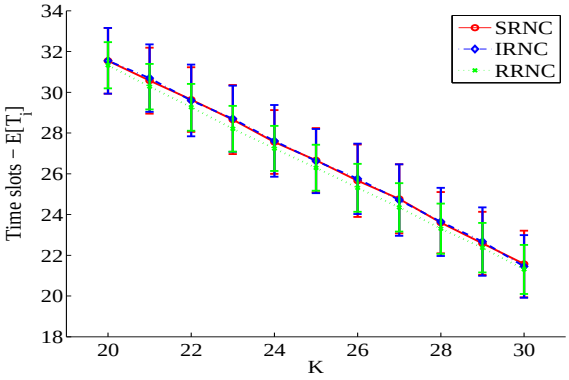

Figure 3: Empirical $\mathbf{E}[T_i]$ vs. K

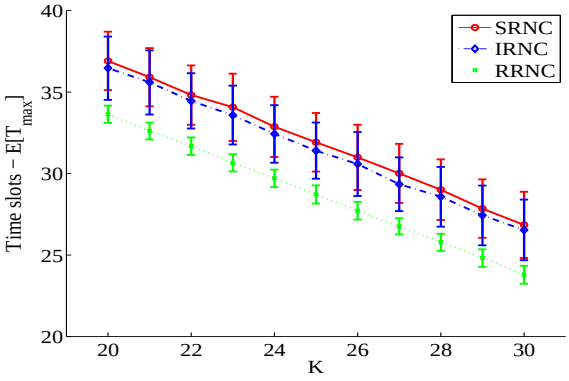

Figure 4: Empirical $\mathbf{E}[T_{max}]$ vs. K

Next, we verify our theoretical results with simulations for various parameters. Using $D = 100, F = 2, K = 30 \rightarrow 70$, Fig. 5 and Fig. 6 show the correctness of analytical performance of the SRNC algorithm. As seen, the number of time slots decreases while K increases. Intuitively, the more information a receiver has, the less information the sender needs to broadcast to complete the synchronization process at this receiver.

Using $N = 10, D = 10, F = 2, K = 5$, Fig. 7 and Fig. 8 verify the agreement between theoretical and simulated performance results of the IRNC algorithm as function of M (cardinality of the union set). As seen, the smaller cardinality of the union set is, the better performance can be achieved.

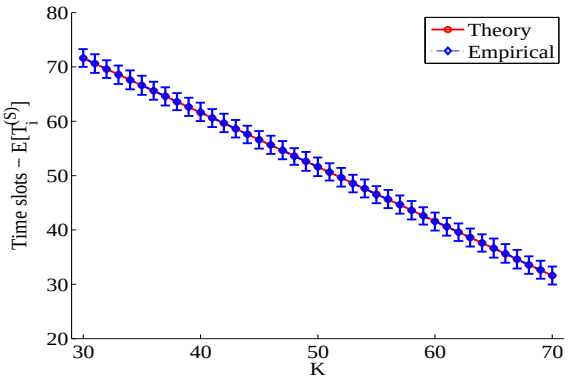

Figure 5: Theoretical and empirical $\mathbf{E}[T_i^{(S)}]$ vs. K for SRNC

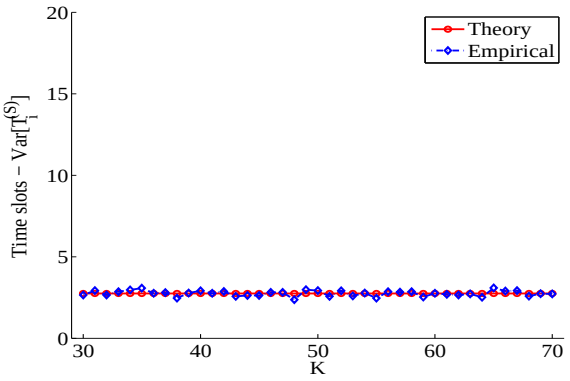

Figure 6: Theoretical and empirical $\mathbf{Var}[T_i^{(S)}]$ vs. K for SRNC

Using $N = 50, D = 30, F = 2, K = 10$, the upper bound on the expectation and the variance of the RRNC algorithm are shown in Fig. 9 and Fig. 10.

Fig. 11 and Fig. 12 show the performance of three algorithms with different value of F (the field size) using $N = 30, D = 20, K = 10, F \in \{2, 3, 5, 7, 11, 13\}$. As seen, while F grows the performance of the proposed algorithms is improved substantially. Intuitively, with a larger field size the probability that a new generated packet is dependent with the packets in "Has" sets at receivers will decrease, leading to higher chance recovering the all original packets faster.

The robustness of random network coding techniques can be verified in Fig. 12. Here, we compare proposed algorithms with an efficient deterministic algorithm in which the sender only broadcasts M original packets in union set $\mathcal{W}$. Obviously, the number of time slots to complete synchronization process for the deterministic algorithm is M. It can be seen that the deterministic algorithm outperforms SRNC and IRNC for some small values of F, however from the range where $F \geq 7$, IRNC has better performance and the performance of SRNC: $\mathbf{E}[T_{max}^{(I)}]$ is very close to M.

We now compare the performance of the RRNC algorithm with the well-known maximum clique network coding algorithm (MCNC) [41]. Since finding maximum clique is an NP-hard problem, the Bron-Kerbosch algorithm is used as the heuristic solution [42].

The pseudo codes for the MCNC algorithm is shown below.

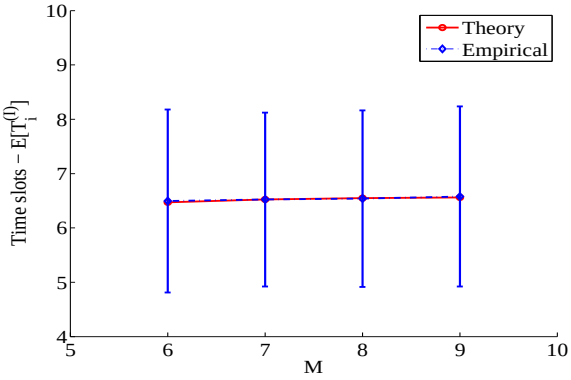

Figure 7: Theoretical and empirical $\mathbf{E}[T_i^{(I)}]$ vs. M for IRNC

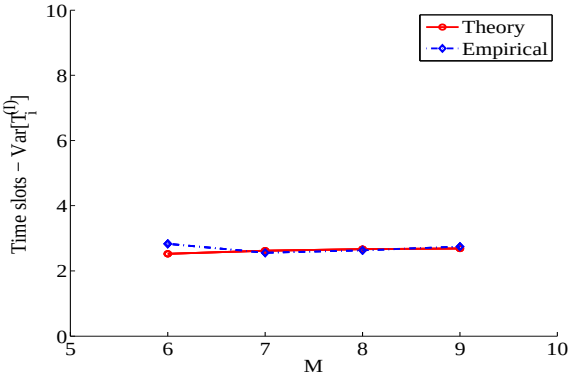

Figure 8: Theoretical and empirical $\mathbf{Var}[T_i^{(I)}]$ vs. M for IRNC

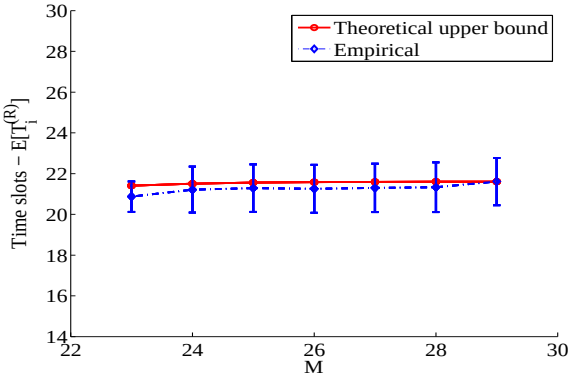

Figure 9: Theoretical upper bound and empirical $\mathbf{E}[T_i^{(R)}]$ vs. M for RRNC

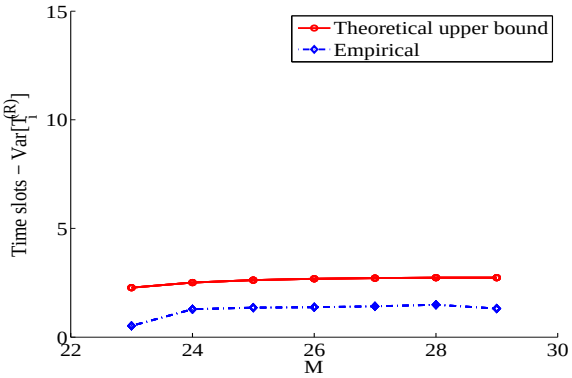

Figure 10: Theoretical upper bound and empirical $\mathbf{Var}[T_i^{(R)}]$ vs. M RRNC

The numbers of rounds vs the size of "Has" set for the MCNC and RRNC algorithms from the sender and receiver's perspectives, are shown in the Fig. 13 and Fig. 14, respectively. In this simulation, we use $N = 20, D = 20, F = 2$.

The shorter the round is the better the performance is. As seen, both algorithms have almost identical performance. At the receiver, both algorithms work in the same manner. However, it should be noted that at the sender the RRNC algorithm is much faster since it does not need to find the maximum clique which has the complexity of $O(3^{KN})/3)$ in the worst case for using Bron-Kerbosch algorithm [43]. Rather, the RRNC algorithm simply uses random projection with the complexity of $O(ND)$ which appears to perform similarly to the MCNC algorithm.

IX. Conclusion

In this paper, we describe the problem of efficient data synchronization/ broadcast for a large number of nodes with disparate data. The synchronization problem arises naturally in many applications, including Peer-to-Peer networks, data centers, and distributed storage systems with asynchronous updates. Two probabilistic models are considered on how the initial fractions of packets at receivers are distributed and according to different practical scenarios. Also, we propose and analyze a number of random network coding algorithms and verify their performances via theoretical analysis and simulations.

Appendix

Proposition 5

Proof. Let $t_j^{(S)}$ be the random variable representing the number of time slots to collect the j-th linearly independent packet after $(j-1)$ linearly independent packets has been added to $\mathcal{H}_i$ (in addition to K linearly independent packets in $\mathcal{H}_i$ at initial). In $\mathcal{H}_i$, there are $(K+j-1)$ linearly independent packets so there are $(F^{K+j-1}-1)$ dependent vectors with $\mathcal{H}_i$ in total of (F^D-1) nonzero vectors in $GF(\mathcal{F}^D)$.

Let $p_j^{(S)}$ be the probability the j-th linearly independent packet is received at each time slot. We have:

$$p_j^{(S)} = 1 - \frac{F^{K+j-1}-1}{F^D-1} = \frac{F^D - F^{K+j-1}}{F^D-1}$$

Then $t_j^{(S)}$ has geometric distribution with expectation $\mathbf{E}[t_j^{(S)}] = \frac{1}{p_j^{(S)}}$ and variance $\mathbf{Var}[t_j^{(S)}] = \frac{1-p_j^{(S)}}{p_j^{(S)2}}$. Since $\mathbf{H}_i$ needs exactly L new linearly independent packets to be full rank, the number of broadcasts $T_i^{(S)}$ that receiver R_i can

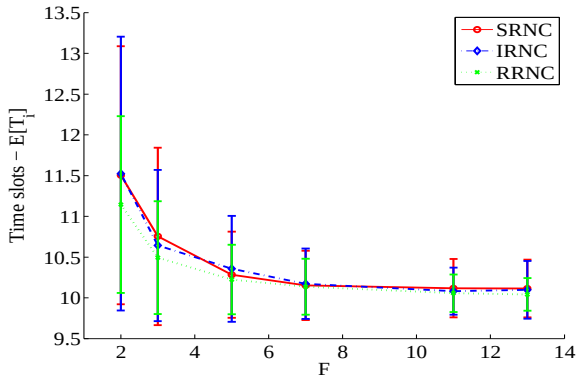

Figure 11: Empirical $\mathbf{E}[T_i]$ of receiver as a function of F

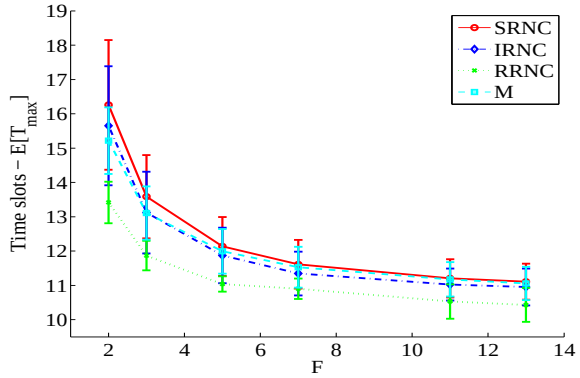

Figure 12: Empirical $\mathbf{E}[T_{max}]$ of sender as a function of F

recover all D original packets is equal the time it receives L-th new linearly independent packet:

$$\mathbf{E}[T_i^{(S)}] = \sum_{j=1}^{L} E[t_j^{(S)}] = \sum_{j=1}^{L} \frac{1}{p_j^{(S)}}$$

$$\rightarrow \mathbf{E}[T_i^{(S)}] = \sum_{j=1}^{L} \frac{F^D - 1}{F^D - F^{K+j-1}} \tag{49}$$

Also, for the variance of $T_i^{(S)}$:

$$\mathbf{Var}[T_i^{(S)}] = \sum_{j=1}^{L} \mathbf{Var}[t_j^{(S)}] = \sum_{j=1}^{L} \frac{1 - p_j^{(S)}}{{p_j^{(S)}}^2}$$

$$\rightarrow \mathbf{Var}[T_i^{(S)}] = \sum_{j=1}^{L} \frac{(F^{K+j-1} - 1)(F^D - 1)}{(F^D - F^{K+j-1})^2} \tag{50}$$

□

Proposition 6

Proof. Consider the behavior at receiver R_i, let $\mathcal{S}_i$ be the intersection (share) set between $\mathcal{H}_i$ and the union set $\mathcal{W}$ at the sender. We have

$$|\mathcal{S}_i| = |\mathcal{W} \cap \mathcal{H}_i| = |\mathcal{W}| + |\mathcal{H}_i| - |\mathcal{P}| = M + K - D = M - L.$$

We use the similar approach as in proof of Proposition 5 except that we randomly choose non-zero vectors in $\mathcal{W}$. Also in $\mathcal{H}_i$,

Algorithm 4: MCNC Algorithm

Data: The sender has knowledge about "Want" sets at receivers at each time slot

while *There exists one receiver that cannot recover all the original packets* **do**
- Based on each "Want" set from each receiver, the sender creates a undirected graph in which each node is represented by a pair of one receiver and one wanted packet;
- If both receivers want the same packet or if two receivers satisfy that each wants the packet that other has then we connect these two corresponding nodes in the graph;
- Sender generates and broadcasts a mixed packet based on the maximum clique found in the graph;
- Each receiver R_i updates its "Has" set and corresponding matrix $\mathbf{H}_i$;
- **if** H_i *is full rank* **then**
  - R_i can recover the original packets and sends acknowledgment to the sender;
- **else**
  - R_i computes and sends its "Want" set to the sender;
- **end**

end

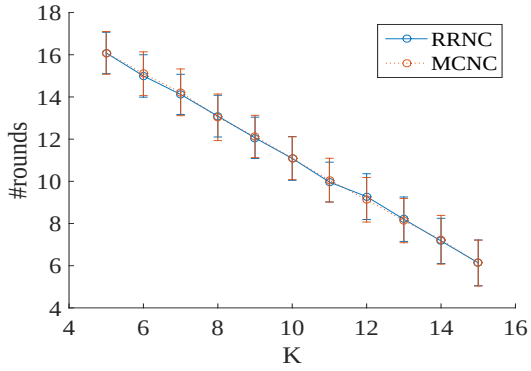

Figure 13: Number of rounds vs. the size of "Has" set for RRNC and MCNC from the receiver's perspective

there are $(M-L)$ packets that are linearly dependent with $\mathcal{W}$. Hence, the probability $p_j^{(I)}$ that the j-th linearly independent packet is received at R_i can be computed as follows.

$$p_j^{(I)} = 1 - \frac{F^{M-L+j-1} - 1}{F^M - 1} = \frac{F^M - F^{M-L+j-1}}{F^M - 1}$$

Now, for the expectation and variance of $T_i^{(I)}$

$$\mathbf{E}[T_i^{(I)}] = \sum_{j=1}^{L} \frac{1}{p_j^{(I)}} = \sum_{j=1}^{L} \frac{F^M - 1}{F^M - F^{M-L+j-1}}. \tag{51}$$

$$\mathbf{Var}[T_i^{(I)}] = \sum_{j=1}^{L} \mathbf{Var}[t_j^{(I)}] = \sum_{j=1}^{L} \frac{1 - p_j^{(I)}}{{p_j^{(I)}}^2}$$

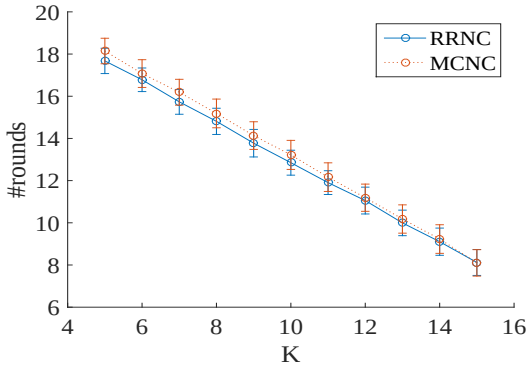

Figure 14: Number of rounds vs. the size of the "Has" set for RRNC and MCNC from the sender's perspective

$$\rightarrow \mathbf{Var}[T_j^{(I)}] = \sum_{j=1}^{L} \frac{(F^{M-L+j-1}-1)(F^M-1)}{(F^M - F^{M-L+j-1})^2}. \quad (52)$$

□

Proposition 7

Proof. After each transmission, every receiver R_i recomputes its "Want" set $\mathcal{W}_i$, and then the sender recomputes $\mathcal{W}$, so the cardinality $M = |\mathcal{W}|$ will decrease by at least one. Let W_j be the updated union set at the sender after R_i receive the $(j-1)$-th linearly independent packets. We have $|\mathcal{W}_j| < |\mathcal{W}_{j-1}| < \cdots < |\mathcal{W}_1| = |\mathcal{W}|$ and $|\mathcal{W}_j| = M_j \leq M-(j-1) = M-j+1$. Now, the intersection (share) set $\mathcal{S}_{i,j}$ between $\mathcal{H}_i$ and the union set $\mathcal{W}_j$ is $\mathcal{S}_{i,j}$. We have

$$\begin{aligned} |\mathcal{S}_{i,j}| &= |\mathcal{W}_j \cap \mathcal{H}_i| = |\mathcal{W}_j| + |\mathcal{W}_i| - |\mathcal{P}| \\ &= M_j + K - D = M_j - L. \end{aligned}$$

Then the probability $p_j^{(R)}$ such that the j-th new linearly independent packet is received can be computed as follows.

$$p_j^{(R)} = 1 - \frac{F^{M_j-L+j-1}-1}{F^{M_j}-1} = \frac{F^{M_j} - F^{M_j-(L-j+1)}}{F^{M_j}-1}$$

Consider the following function:

$$f(x) = \frac{F^x - F^{x-a}}{F^x - 1}$$

where $a = L - j + 1$ then $1 \leq a \leq L$. We have:

$$f'(x) = -\frac{(F^a-1)\ln(F)F^{x-a}}{(F^x-1)^2} \leq 0.$$

Hence, $f(x)$ is monotonically decreasing. Since $M_j \leq M - j + 1$, we have:

$$p_j^{(R)} = f(M_j) \geq f(M-j+1) = \frac{F^{M-j+1} - F^{M-L}}{F^{M-j+1}-1} \quad (53)$$

Therefore,

$$\mathbf{E}[T_i^{(R)}] = \sum_{j=1}^{L} \frac{1}{p_j^{(R)}} \leq \sum_{j=1}^{L} \frac{F^{M-j+1}-1}{F^{M-j+1} - F^{M-L}} \quad (54)$$

For the variance of $T_i^{(I)}$, we have

$$\mathbf{Var}[T_i^{(R)}] = \sum_{j=1}^{L} \frac{1-p_j^{(R)}}{{p_j^{(R)}}^2} \quad (55)$$

Consider the following function

$$g(x) = \frac{1-x}{x^2}$$

We have

$$g'(x) = \frac{x-2}{x^3} < 0$$

where $0 \leq x \leq 1$. Hence, $g(x)$ is a monotonically decreasing function in $0 \leq x \leq 1$. Combine with (53), we have:

$$\mathbf{Var}[T_i^{(R)}] = \sum_{j=1}^{L} g(p_j^{(R)}) \leq \sum_{j=1}^{L} \frac{(F^{M-j+1}-1)(F^{M-L}-1)}{(F^{M-j+1} - F^{M-L})^2}.$$

□

Proposition 9

Proof. The proof approaches are similar for all three algorithms. Here, the general notation T_{max} can be applied to each algorithm, respectively. We have

$$\mathbf{P}(T_{max} > a) = 1 - \mathbf{P}(T \leq a).$$

Also, $\mathbf{P}(T_{max} \leq a) = \mathbf{P}(\bigcap_{i=1}^{N} T_i \leq a)$.

Since $a > \mu$, let $a = b\sigma + \mu$ where $b > 0$ then

$$\begin{aligned} \mathbf{P}(T \leq a) &= \mathbf{P}(\bigcap_{i=1}^{N} T_i \leq \mu + b\sigma) \\ &= \mathbf{P}(\bigcap_{i=1}^{N} T_i - \mu \leq b\sigma) \\ &\geq \mathbf{P}(\bigcap_{i=1}^{N} |T_i - \mu| \leq b\sigma) \end{aligned}$$

Apply two-sided Chebyshev's inequality with N independent random variables $T_1, T_2, \ldots, T_N$:

$$\mathbf{P}(\bigcap_{i=1}^{N} |T_i - \mu| \leq b\sigma) \geq \prod_{i=1}^{N} (1 - \frac{1}{b^2}) = (1 - \frac{1}{b^2})^N$$

Note: the bound is only meaningful where $b \geq 1$.
Hence,

$$\mathbf{P}(T > a) \leq 1 - (1 - \frac{1}{b^2})^N$$

Plug $b = \frac{a-\mu}{\sigma}$ back, we have:

$$\mathbf{P}(T > a) \leq 1 - (1 - \frac{\sigma^2}{(a-\mu)^2})^N \quad (56)$$

□

REFERENCES

[1] Y. Cui, H. Wang, X. Cheng, and B. Chen, "Wireless data center networking," *Wireless Communications, IEEE*, vol. 18, no. 6, pp. 46–53, December 2011.

[2] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, "Augmenting data center networks with multi-gigabit wireless links," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 38–49.

[3] Y. Cui, H. Wang, and X. Cheng, "Wireless link scheduling for data center networks," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*. ACM, 2011, p. 44.

[4] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1479–1494, March 2011.

[5] Y. Birk and T. Kol, "Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2825–2830, 2006.

[6] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 244–248.

[7] H. Y. Kwan, K. W. Shum, and C. W. Sung, "Generation of innovative and sparse encoding vectors for broadcast systems with feedback," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1161–1165.

[8] K. Chi, X. Jiang, and S. Horiguchi, "Network coding-based reliable multicast in wireless networks," *Computer Networks*, vol. 54, no. 11, pp. 1823–1836, 2010.

[9] N. J. Harvey, D. R. Karger, and S. Yekhanin, "The complexity of matrix completion," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 1103–1111.

[10] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2004, pp. 142–150.

[11] R. Peeters, "Orthogonal representations over finite fields and the chromatic number of graphs," *Combinatorica*, vol. 16, no. 3, pp. 417–431, 1996.

[12] S. H. Dau, V. Skachek, and Y. M. Chee, "Optimal index codes with near-extreme rates," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1515–1527, 2014.

[13] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Information Theory Workshop, 2007. ITW'07. IEEE*. IEEE, 2007, pp. 120–125.

[14] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*. IEEE, 2008, pp. 315–319.

[15] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," *arXiv preprint arXiv:1004.1379*, 2010.

[16] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *Information Theory, IEEE Transactions on*, vol. 54, no. 12, pp. 5511–5524, Dec 2008.

[17] M. A. R. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *INFOCOM Workshops 2008, IEEE*, April 2008, pp. 1–4.

[18] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, pp. 2235–2245.

[19] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, p. 77, 2007.

[20] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proceedings of the annual Allerton conference on communication control and computing*, vol. 41, no. 1. The University; 1998, 2003, pp. 40–49.

[21] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, pp. 497–510, 2008.

[22] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *Information Theory, IEEE Transactions on*, vol. 56, no. 7, pp. 3187–3195, July 2010.

[23] S. E. Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," *arXiv:1002.1465 [cs, math]*, Feb. 2010, 00038. [Online]. Available: http://arxiv.org/abs/1002.1465

[24] A. Sprintson, P. Sadeghi, G. Booker, and S. E. Rouayheb, "Deterministic algorithm for coded cooperative data exchange," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, X. Zhang and D. Qiao, Eds. Springer Berlin Heidelberg, Jan. 2012, no. 74, pp. 282–289, 00011.

[25] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 1888–1892.

[26] T. A. Courtade and R. D. Wesel, "Coded cooperative data exchange in multihop networks," *arXiv:1203.3445 [cs, math]*, Mar. 2012, 00006. [Online]. Available: http://arxiv.org/abs/1203.3445

[27] M. Yan and A. Sprintson, "Algorithms for weakly secure data exchange," in *Network Coding (NetCod), 2013 International Symposium on*, June 2013, pp. 1–6.

[28] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 2, pp. 914–925, Feb 2009.

[29] T. Tran, T. Nguyen, B. Bose, and V. Gopal, "A hybrid network coding technique for single-hop wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 685–698, 2009.

[30] J. Liu, D. Goeckel, and D. Towsley, "The throughput order of ad hoc networks employing network coding and broadcasting," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Oct 2006, pp. 1–7.

[31] Y. E. Sagduyu and A. Ephremides, "On joint mac and network coding in wireless ad hoc networks," *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3697–3713, 2007.

[32] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 401–412. [Online]. Available: http://doi.acm.org/10.1145/1402958.1403004

[33] A. Al Hamra, C. Barakat, and T. Turletti, "Network coding for wireless mesh networks: A case study," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*. IEEE Computer Society, 2006, pp. 103–114.

[34] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[35] R. Koetter and M. Médard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, 2003.

[36] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[37] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 41, no. 1. The University; 1998, 2003, pp. 11–20.

[38] H. A. David and H. N. Nagaraja, *Order statistics*. Wiley Online Library, 1970.

[39] B. C. Arnold and R. A. Groeneveld, "Bounds on expectations of linear systematic statistics based on dependent samples," *The Annals of Statistics*, vol. 7, no. 1, pp. 220–223, 01 1979. [Online]. Available: http://dx.doi.org/10.1214/aos/1176344567

[40] C. Cooper, "On the distribution of rank of a random matrix over a finite field," *Random Structures and Algorithms*, vol. 17, no. 3-4, pp. 197–212, 2000.

[41] X. Wang, J. Wang, and Y. Xu, "Data dissemination in wireless sensor networks with network coding," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 1, p. 1, 2010.

[42] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[43] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.