

On the Packet Decoding Delay of Linear Network Coded Wireless Broadcast

Mingchao Yu, Alex Sprintson[†], and Parastoo Sadeghi*

*Research School of Engineering, Australian National University, Canberra, Australia

[†]Department of Electrical and Computer Engineering, Texas A&M University, Texas, USA

Emails: {ming.yu, parastoo.sadeghi}@anu.edu.au, spalex@tamu.edu

Abstract

We apply linear network coding (LNC) to broadcast a block of data packets from one sender to a set of receivers via lossy wireless channels, assuming each receiver already possesses a subset of these packets and wants the rest. We aim to characterize the average packet decoding delay (APDD), which reflects how soon each individual data packet can be decoded by each receiver on average, and to minimize it while achieving optimal throughput. To this end, we first derive closed-form lower bounds on the expected APDD of all LNC techniques under random packet erasures. We then prove that these bounds are NP-hard to achieve and, thus, that APDD minimization is an NP-hard problem. We then study the performance of some existing LNC techniques, including random linear network coding (RLNC) and instantly decodable network coding (IDNC). We proved that all throughput-optimal LNC techniques can approximate the minimum expected APDD with a ratio between $4/3$ and 2 . In particular, the ratio of RLNC is exactly 2 . We then prove that all IDNC techniques are only heuristics in terms of throughput optimization and cannot guarantee an APDD approximation ratio for at least a subset of the receivers. Finally, we propose hyper-graphic linear network coding (HLNC), a novel throughput-optimal and APDD-approximating LNC technique based on a hypergraph model of receivers' packet reception state. We implement it under different availability of receiver feedback, and numerically compare its performance with RLNC and a heuristic general IDNC technique. The results show that the APDD performance of HLNC is better under all tested system settings, even if receiver feedback is only collected intermittently.

Keywords: *wireless broadcast, linear network coding, delay, combinatorial optimization, hypergraph coloring*

I. INTRODUCTION

A. Background

In this paper, we consider a wireless broadcast problem where a sender wishes to broadcast a block of K data packets to a set of N receivers using linear network coding (LNC) [1]–[5]. Each receiver is assumed to already possess a subset of the data packets and still wants all the remaining data packets. The sender transmits, as LNC coded packets, linear combinations of the data packets, so that the receivers can decode their missing data packets through solving linear equations. We are interested in studying two interrelated performance metrics, namely throughput and average packet decoding delay (APDD) across the receivers: Throughput measures the time it takes to deliver the whole data block to all receivers. APDD measures the average time it takes each receiver to decode each data packets in the block. For example:

Example 1. Consider two receivers, r_1 and r_2 , and a block of three data packets $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. r_1 already has \mathbf{p}_1 and r_2 already has \mathbf{p}_2 . They both want the remaining two data packets. Compare two LNC schemes:

- Scheme-1: send $\mathbf{p}_1 + \mathbf{p}_2$, and then \mathbf{p}_3 ;
- Scheme-2: send $\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3$, and then $\mathbf{p}_1 + 2\mathbf{p}_2 + 3\mathbf{p}_3$.

Both schemes allow the two receivers to fully decode after two transmissions. Thus, they offer the same throughput. However, they offer different APDD: Scheme-1 allows both receivers to decode one data packet after both the first and second transmissions, yielding an APDD of $\frac{1+1+2+2}{4} = 1.5$. But Scheme-2 does not allow any packet decodings after the first transmission, yielding a larger APDD of $\frac{2+2+2+2}{4} = 2$.

A lower APDD implies faster and smoother data delivery to the application layer within each data block, and is particularly important in applications where individual data packets are informative. Thus, we are interested in the the minimization of APDD and its relation to throughput. However, as we will review next, APDD minimization is an open problem in the literature.

B. Some Existing LNC Classes

A well-known class of LNC techniques, which will be studied in this paper, is the class of throughput-optimal LNC techniques, whose packets are innovative (bring new information) to every receiver who has not yet fully decoded all the packets in the block. Throughput-optimal packets can be generated either randomly (i.e., the classic random LNC (RLNC) technique [4], [6], [7]) or deterministically (e.g., by solving a hitting set problem [8], or by solving a matroid graphic representation problem [9]). All these throughput-optimal techniques could suffer from large APDD due to block decoding: each receiver is only able to decode all data packets at once after receiving sufficient linearly independent coded packets. In other words, there is generally no early packet decodings, which would help reduce APDD. In order to provide early packet decodings without sacrificing throughput, Keller *et al* proposed in [10] a two-step coding technique that adds extra data packets to an instantly decodable but throughput-suboptimal coded packet for throughput optimality. But this technique was not analytically studied. Indeed, the APDD of throughput-optimal LNC techniques has not been well studied. Only recently, Yu *et al* proved in [11] that, when there is no packet erasure, RLNC is an APDD 2-approximation technique. That is, its APDD is at most twice of the minimum possible APDD that any LNC techniques can offer.

Another well-known class of LNC techniques, which will also be studied in this paper, is instantly decodable network coding (IDNC). This class aims at reducing APDD through early packet decodings [12]–[16]. The approach is to send, as coded packets, binary XORs of selected data packets, so that a subset of (or all) receivers can instantly decode one wanted data packet from each coded packet. It is well known that IDNC techniques are generally not throughput optimal [17], [18]. Moreover, [15] proved that it is intractable to maximize the throughput subject to IDNC constraints. This suboptimal throughput, in turn, can hurt APDD, for it brings large delays to data packets decoded in the final stage of the broadcast. Although a large body of heuristics have been developed as a remedy [15], [19]–[21], it remains an open problem whether or not IDNC techniques are able to at least approximate the optimal throughput and APDD. Only recently, Yu *et al* proved in [11] that the class of strict IDNC techniques [12], [14], [16] are not APDD-approximation techniques even when there are no packet erasures.

There are also LNC techniques that strike a balance between throughput and APDD [22]–[24]: They

partition the packet block into sub-blocks and broadcast each sub-block separately, so that data packets from earlier sub-blocks can be decoded sooner. However, these techniques are generally heuristic. Thus, they will not be studied in this paper.

We also note that index coding (IC) considers a more general system setting than this paper, which assumes that every receiver has a subset of the K data packets and wants one [25]–[28] or some [29] of the rest. While throughput optimization has been extensively studied in the IC literature, APDD minimization has not previously been considered in the IC literature to the best of our knowledge. Moreover, most works in the IC literature assume no packet erasures. Therefore, our study may also provide new insights into the corresponding problem in the IC context.

C. Contributions

In summary, APDD minimization for erasure-prone LNC wireless broadcast systems is a highly nontrivial problem. There have not been any comprehensive studies on the APDD performance of LNC techniques, nor any optimal or approximation techniques, only heuristics. We are thus motivated to close this knowledge gap. Specifically, in this paper we achieve the following main contributions:

- 1) We reveal the APDD performance limits of LNC techniques by deriving closed-form lower bounds on the expected APDD¹ of LNC techniques under random packet erasures;
- 2) We prove that APDD minimization is NP-hard;
- 3) We prove that RLNC is an APDD 2-approximate technique. We also prove that all throughput-optimal LNC techniques can approximate the minimum expected APDD with a ratio between $4/3$ and 2 ;
- 4) We prove that all IDNC techniques cannot approximate the optimal throughput, and show that they cannot guarantee an APDD approximation ratio for at least a subset of the receivers;
- 5) We propose hypergraphic linear network coding (HLNC), a novel throughput-optimal and APDD-approximation LNC technique built upon a novel hypergraph representation of receivers' knowledge space. Extensive simulations show that it outperforms RLNC and a heuristic general IDNC

¹By “average” packet decoding delay, we mean the average of the decoding delay of every data packet at every receiver in a given instance of the problem. By “expected” APDD, we mean the statistical average of the APDD of all possible instances of the problem generated under random packet erasures.

technique in terms of APDD under all tested system parameter settings. HLNC is implementation-friendly, for it does not require NP-hard coding decision making nor receiver feedback after every transmission.

The rest of this paper is organized as follows:

- Section 2 defines the system model, notations, and terminologies;
- Section 3 studies the fundamental features of APDD, including lower bounds on the expected APDD, and the NP-hardness of the APDD minimization problem;
- Section 4 studies the performance of RLNC and IDNC techniques;
- Section 5 proposes HLNC and discusses its implementations under different availability of receiver feedback;
- Section 6 numerically compares the APDD performance of HLNC with RLNC and a general IDNC technique. It also numerically demonstrates the feedback reduction capability of HLNC;
- Section 7 concludes the paper.

Remark 1. *We note that preliminary results of some of the above contributions have been derived in [11] under the limited setting without packet erasures. As we will see later, their extension to settings with packet erasures is non-trivial, and requires completely different derivation approaches.*

II. SYSTEM MODEL

A. System Setting

We consider a block-based wireless broadcast scenario, in which the sender wishes to deliver a block of K data packets, denoted by $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^K$, to a set of N receivers, denoted by $\mathcal{R} = \{r_n\}_{n=1}^N$. All data packets are vectors of the same length, with entries taken from a finite field \mathbb{F}_q . Time is slotted, and in each time slot the sender broadcasts a data or coded packet to all receivers. The wireless channel between the sender and each receiver r_n is independent of each other, and is subject to Bernoulli random packet erasures with a probability of $P_{e,n} \geq 0$.

We assume that each receiver has already received a subset of packets in \mathcal{P} , and still wants all the rest. Such a packet reception state could be the consequence of previous uncoded transmissions [7],

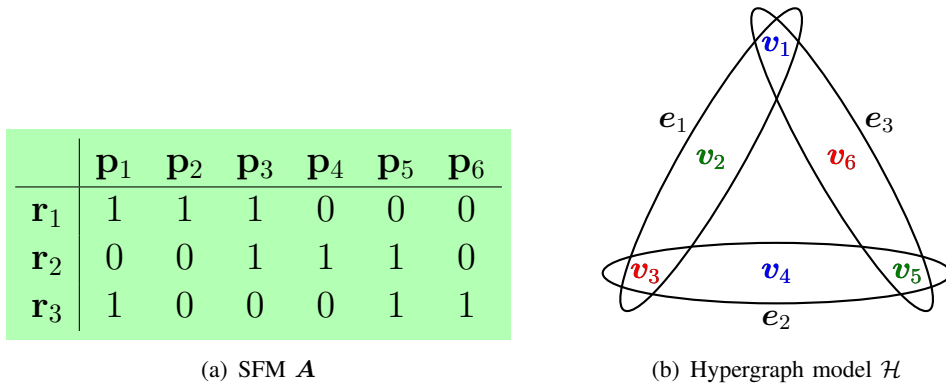


Fig. 1: An example of state feedback matrix \mathbf{A} and its hypergraph representation \mathcal{H} . Vertices in \mathcal{H} are strong-colored, such that all vertices in the same hyperedge have different colors.

and is a common assumption in network coding and index coding literature [25]–[29]. This state can be summarized by a binary $N \times K$ state feedback matrix (SFM) \mathbf{A} . Here $\mathbf{A}(n, k) = 1$ means r_n has missed \mathbf{p}_k and still wants it, and $\mathbf{A}(n, k) = 0$ means r_n already has \mathbf{p}_k . The set of data packets wanted by r_n is denoted by \mathcal{W}_n . Its size is denoted by w_n . An example of \mathbf{A} is given in Fig. 1(a). There are 6 data packets and 3 receivers. Receivers $\{r_1, r_2, r_3\}$ all want 3 data packets.

The sender then applies an LNC technique to help receivers recover their missing data packets. In each time slot, it broadcasts an LNC packet \mathbf{X} , which takes the form of:

$$\mathbf{X} = \sum_{\mathbf{p}_k \in \mathcal{M}} \alpha_k \mathbf{p}_k, \quad (1)$$

where \mathcal{M} is a subset of \mathcal{P} and is called the coding set of \mathbf{X} , and $\{\alpha_k\}$ are non-zero coding coefficients chosen from \mathbb{F}_q , i.e., $\{\alpha_k\} \subseteq \mathbb{F}_q \setminus \{0\}$. Similarly, \mathbf{X} is called a coded packet of \mathcal{M} . Moreover, if $\{\alpha_k\}$ are chosen uniformly at random from $\mathbb{F}_q \setminus \{0\}$, then \mathbf{X} is called a random-coded packet of \mathcal{M} . When \mathbb{F}_q is sufficiently large, any receiver who is missing k data packets in \mathcal{M} can decode them from a set of k random-coded packets of \mathcal{M} w.h.p.

B. Definitions and Terminologies

Average packet decoding delay (APDD) measures how fast each data packet is decoded on average. Given an instance of the broadcast, the APDD of receiver r_n is the average time it takes for r_n to decode

a data packet, and is denoted by D_n :

$$D_n = \frac{1}{w_n} \sum_{\forall k: \mathbf{p}_k \in \mathcal{W}_n} u_{n,k}, \quad (2)$$

where $u_{n,k}$ is the index of the packet transmission after which r_n decodes \mathbf{p}_k . The APDD across all receivers is similarly defined as

$$D = \frac{1}{\text{sum}(\mathbf{A})} \sum_{\forall k, n: \mathbf{A}(n,k)=1} u_{n,k}, \quad (3)$$

where $\text{sum}(\mathbf{A})$ is the sum of the entries of \mathbf{A} , and is equal to the number of ones in \mathbf{A} . The expected APDD $E[D_n]$ of a receiver (resp. the expected APDD $E[D]$ across all receivers) under random packet erasures can then be derived by averaging D_n (resp. D) over all possible packet erasure patterns.

We further denote by U_n the number of transmissions after which receiver r_n decodes all its wanted data packets. Then, $U \triangleq \max_{n \in [1, N]} U_n$ is the block completion time (BCT). We use BCT to measure system throughput, as throughput is inversely proportional to BCT. The expected U_n and U under random packet erasures are denoted by $E[U_n]$ and $E[U]$, and can be derived by averaging U_n and U over all possible packet erasure patterns.

We define an APDD β -approximation LNC technique as follows:

Definition 1. *An LNC technique is an APDD β -approximation technique if for any given SFM \mathbf{A} and packet erasure probabilities $\{P_{e,n}\}_{n=1}^N$, the expected APDD $E[D]$ of using this technique is at most β times of the minimum expected APDD $E[D_{\min}]$ that any LNC techniques can offer.*

Similarly, we define an throughput-approximation LNC techniques as follows:

Definition 2. *An LNC technique is a throughput β -approximation technique if for any given SFM \mathbf{A} and packet erasure probabilities $\{P_{e,n}\}_{n=1}^N$, the expected BCT $E[U]$ of using this technique is at most β times of the minimum expected BCT $E[U_{\min}]$ that any LNC techniques can offer.*

III. FUNDAMENTAL APDD PERFORMNACE LIMITS

In this section, we will derive lower bounds on the expected APDD of LNC techniques. We will then prove that such lower bounds are NP-hard to achieve, which will indicate that APDD minimization is an NP-hard problem.

Our approach is to investigate the performance and existence of a “perfect LNC technique”:

Definition 3. *A LNC technique is perfect if it allows every receiver who is still missing packet(s) to decode one new wanted data packet whenever this receiver successfully receives a coded packet generated using this technique.*

For example, for the A in Fig. 1(a), a perfect LNC technique could send 3 coded packets when there are no packet erasures: $X_1 = p_1 \oplus p_4$, $X_2 = p_2 \oplus p_5$, and $X_3 = p_3 \oplus p_6$, where \oplus is the binary XOR operation. $\{X_u\}_{u=1}^3$ allow every receiver to decode one wanted data packet in every transmission. For example, r_1 can decode p_1 from X_1 by performing $X_1 \oplus p_4$, as it already has p_4 .

A. Lower Bound on The Expected APDD

It is intuitive that the APDD of the perfect LNC technique is a lower bound on the APDD of LNC techniques. We thus denote by \underline{D}_n (resp. \underline{D}) the APDD of receiver r_n (resp. across all receiver) using the perfect LNC technique. Their expectations under random packet erasures, namely, $E[\underline{D}_n]$ and $E[\underline{D}]$, are thus the lower bounds on the corresponding expectations across all LNC techniques.

The value of $E[\underline{D}_n]$ is stated in the following theorem:

Theorem 1. *When coded transmissions are subject to random packet erasures with a probability of $P_{e,n}$, the expected APDD of a receiver r_n who wants w_n data packets is lower bounded as:*

$$E[\underline{D}_n | \text{when } r_n \text{ wants } w_n \text{ data packets}] = \frac{w_n + 1}{2(1 - P_{e,n})} \quad (4)$$

The proof is given in Appendix B. In the rest of the paper, we will simplify the notation for such conditional expectations to a form of $E[A|(b, c, \dots)]$, which stands for the expectation of random variable

A when the values of some random variables are given as b, c, \dots . For example, the expectation in the above equation can be simplified to $E[\underline{D}_n|w_n]$.

Then, by averaging the APDD of all receivers with their w_n as weights, we obtain a lower bound on the expected APDD D of any A under any $\{P_{e,n}\}$:

Corollary 1. *The expected APDD of a wireless broadcast instance with given A and $\{P_{e,n}\}$ is lower bounded as:*

$$E[\underline{D}|\{w_n\}_{n=1}^N] = \frac{\sum_{n=1}^N \frac{w_n^2 + w_n}{2(1-P_{e,n})}}{\sum_{n=1}^N w_n}, \quad (5)$$

where $\{w_n\}$ is obtained from A .

Then, by assuming that A is the result of one round of uncoded transmission of the K data packets and that all receivers experience the same packet erasure probability P_e , we obtain a closed-form relation between the APDD performance limit of LNC techniques and system parameters:

Theorem 2. *Given system parameters K, N , and P_e , the overall APDD performance of every NC technique is lower bounded by $E[\underline{D}]$, where*

$$E[\underline{D}] = \frac{1}{2(1-P_e)} \left(1 + E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] \right), \quad \{w_n\}_{n=1}^N \sim B(K, P_e) \quad (6)$$

$$\approx \frac{KP_e - P_e + 2}{2 - 2P_e}, \quad \text{when } N \text{ is sufficiently large} \quad (7)$$

Here each w_n follows a binomial distribution of $B(K, P_e)$ because in the uncoded transmission round r_n will miss each data packet with a probability of P_e . (6) is a straightforward extension of (5). We will prove in Appendix C that $E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] \approx KP_e - P_e + 1$ when N is sufficiently large, which will prove the approximation in (7).

Remark 2. *Under the more general setting with heterogeneous $\{P_{e,n}\}$, we can apply the smallest (resp. largest) $P_{e,n}$ to (7) for a lower (resp. upper) bound on the expected APDD of the perfect LNC technique.*

To verify the accuracy of the approximation in (7), we simulate the perfect LNC technique as follows: We first send each data packet uncoded once. Then in each time slot, instead of finding a perfect coded

packet, we simply give each unfinished receiver one new wanted data packet if its channel is “on”. The APDD of this simulated technique is thus \underline{D} . We have conducted extensive simulations to obtain the numerical average of \underline{D} , and compared it with (7). The results under $K = 15$ and $P_e = 0.2$ are presented in Fig. 2, which shows that our approximation quickly converges to the numerical average when N is only 20. Another interesting observation from both the graph and (7) is that the APDD of the perfect LNC technique does not increase with the number N of receivers.

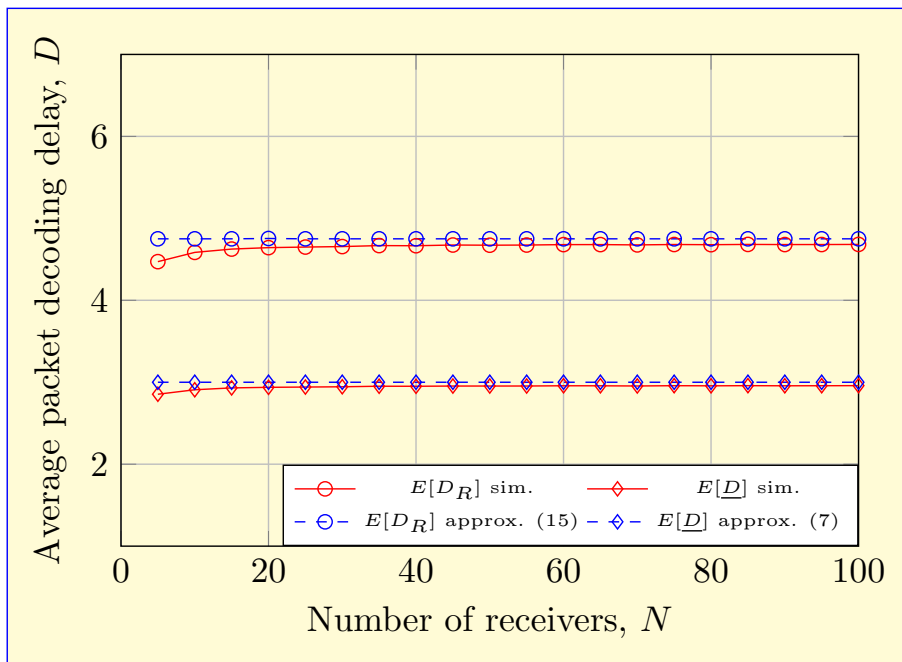


Fig. 2: The proposed closed-form approximation of $E[\underline{D}]$ and $E[D_R]$ and their comparison to the simulated values.

B. The Hardness of Minimizing the Expected APDD

In this section, we prove the NP-hardness of minimizing the expected APDD of a given \mathbf{A} , i.e., $E[D]$. Our approach is through contradiction: If there exists an efficient coding algorithm that minimizes $E[D]$, then this algorithm can efficiently determine the achievability of $E[\underline{D}]$ as well by simply comparing the two quantities. Moreover, such efficiency will hold for the special case where $\{P_{e,n}\} = 0$, which will reduce $E[D]$ and $E[\underline{D}]$ to D and \underline{D} , respectively. Thus, the achievability of \underline{D} can be efficiently determined. Therefore, to prove that it is hard to minimize $E[D]$, it suffices to prove that it is hard to determine the achievability of \underline{D} when $\{P_{e,n}\} = 0$.

To this end, we define a perfect LNC solution, denoted by \mathcal{S}_p , as a set of ordered LNC coded packets that allows every receiver r_n to decode a new wanted data packet in every transmission, until r_n has

decoded all its wanted data packets. To avoid ambiguity, the coded packets do not contain data packets that have already been decoded by all receivers.

It is clear that \underline{D} can only be achieved by \mathcal{S}_p . Thus, to prove that it is hard to determine the achievability of \underline{D} when $\{P_{e,n}\} = 0$, it suffices to prove that it is hard to determine the existence of \mathcal{S}_p when $\{P_{e,n}\} = 0$. We now prove this through a reduction to a hypergraph strong coloring problem.

A hypergraph \mathcal{H} is defined by a pair $(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of vertices, and \mathcal{E} is the set of hyperedges. Every hyperedge $e \in \mathcal{E}$ is a subset of \mathcal{V} with size $|e| \geq 1$. A hypergraph is r -uniform if every hyperedge $e \in \mathcal{E}$ has a size of r . A size- k strong coloring of \mathcal{H} is a partition of \mathcal{V} into k subsets $\{\mathcal{V}_i\}_{i=1}^k$, such that $|\mathcal{V}_i \cap e| \leq 1$ for any $e \in \mathcal{E}$. In other words, if we assign k different colors to the vertices in $\{\mathcal{V}_i\}_{i=1}^k$ respectively, every color appears at most once in every hyperedge. We prove in Appendix A that the strong hypergraph coloring problem is hard:

Lemma 1. *It is NP-complete to determine whether an r -uniform hypergraph is size- r strong colorable or not, for any $r \geq 3$.*

We then build a reduction from the problem of finding a size- r strong coloring for an r -uniform hypergraph to the problem of finding a perfect solution for a given instance of SFM \mathbf{A} . Given an r -uniform hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$ we construct an instance of our problem as follows. For each vertex v_k we introduce a data packet \mathbf{p}_k , and for each hyperedge e_n we introduce a receiver r_n that wants the data packets/vertices incident to e_n . In the resulting SFM \mathbf{A} , every receiver wants r data packets. A 3-uniform hypergraph \mathcal{H} and the corresponding SFM \mathbf{A} are depicted in Fig. 1.

Based on this construction, we can prove the hardness of finding a perfect solution:

Theorem 3. *It is NP-complete to determine whether there exists a perfect solution for a given instance \mathbf{A} of the APDD minimization problem when there are no packet erasures.*

Proof. First, we prove that a size- r strong coloring $\{\mathcal{V}_i\}_{i=1}^r$ of \mathcal{H} implies a perfect solution \mathcal{S}_p for \mathbf{A} in our problem. Since for every hyperedge it holds that $|e_n| = r$ and there are r colors, we have $|\mathcal{V}_i \cap e_n| = 1$. Let $\{\mathcal{M}_i\}_{i=1}^r$ be the sets of packets corresponding to $\{\mathcal{V}_i\}_{i=1}^r$. Then, we have $|\mathcal{M}_i \cap \mathcal{W}_n| = 1$ for every receiver r_n . Hence, the sum (e.g., binary XOR) of all data packets from \mathcal{M}_i is a coded packet, denoted

by \mathbf{X}_i , that allows every receiver to decode a wanted data packet. Therefore, $\{\mathbf{X}_i\}_{i=1}^r$ together form a perfect solution \mathcal{S}_p to our problem.

Next, we prove that a perfect solution \mathcal{S}_p for the \mathbf{A} in our problem implies a size- r strong coloring $\{\mathcal{V}_i\}_{i=1}^r$ of \mathcal{H} . Since every receiver wants r data packets, \mathcal{S}_p contains r network coded packets $\{\mathbf{X}_i\}_{i=1}^r$. In order to allow every receiver to decode one wanted data packet from \mathbf{X}_i , the coding set \mathcal{M}_i of \mathbf{X}_i must contain exactly one wanted data packet of every receiver, i.e., $|\mathcal{M}_i \cap \mathcal{W}_n| = 1$. Let $\{\mathcal{V}_i\}_{i=1}^r$ be the sets of vertices corresponding to $\{\mathcal{M}_i\}_{i=1}^r$. Then it holds that $|\mathcal{V}_i \cap e_n| = 1$ for every hyperedge. Thus, $\{\mathcal{V}_i\}_{i=1}^r$ is a size- r strong coloring of \mathcal{H} .

Our construction above indicates that an r -uniform hypergraph is size- r strong colorable if and only if there exists a perfect solution of the instance \mathbf{A} of our problem. This result, together with Lemma 1, prove the NP-hardness of determining the existence of \mathcal{S}_p . \square

Since \underline{D} can only be achieved by a perfect solution, the above theorem immediately yields the following corollary:

Corollary 2. *It is NP-complete to determine whether \underline{D} is achievable for a given instance \mathbf{A} of the APDD minimization problem when there are no packet erasures.*

As we have discussed at the beginning of this section, a coding algorithm that can efficiently minimize $E[D]$ can also efficiently determine the achievability of \underline{D} when there are no packet erasures. This relation, together with the above corollary, yields the hardness of APDD minimization problem:

Theorem 4. *It is NP-hard to minimize the expected APDD of a given \mathbf{A} when coded transmissions are subject to random packet erasures.*

Given the NP-hardness of APDD minimization, it becomes critical to investigate whether APDD can be approximation or not, as well as its impacts on throughput optimality. To this end, we study two well known classes of LNC techniques that aim at throughput optimization and APDD reduction, respectively.

IV. EXISTING TECHNIQUES

In this section, we study the throughput and APDD performance of two classes of LNC techniques, including RLNC, which is throughput optimal, and IDNC, which aims at APDD reduction. We will derive the expected APDD of RLNC, and use the results to prove that RLNC is an APDD 2-approximation technique. On the other hand, we will prove that IDNC techniques cannot approximate throughput, and can not guarantee an APDD approximation ratio for at least a subset of the receivers.

A. RLNC

Since RLNC is throughput optimal and requires block decoding, a receiver who wants w_n data packets is able to decode them altogether after receiving w_n coded packets, which is expected to take $\frac{w_n}{1-P_e}$ coded transmissions. Hence, the expected APDD of r_n under RLNC is:

$$E[D_{R,n}|w_n] = \frac{w_n}{1 - P_{e,n}} \quad (8)$$

where the ‘‘R’’ in the subscript of $D_{R,n}$ stands for RLNC.

Averaging $E[D_{R,n}|w_n]$ over all receivers yields the expected APDD of a given \mathbf{A} under RLNC:

$$\begin{aligned} E[D_R|\{w_n\}_{n=1}^N] &= \frac{E[D_{R,1}|w_1]w_1 + E[D_{R,2}|w_2]w_2 + \cdots + E[D_{R,N}|w_N]}{w_1 + w_2 + \cdots + w_N} \\ &= \frac{\sum_{n=1}^N \frac{w_n^2}{1-P_{e,n}}}{\sum_{n=1}^N w_n} \end{aligned} \quad (9)$$

Then, by assuming that \mathbf{A} is the consequence of one round of uncoded transmissions of the K data packets, we obtain the following theorem on the overall expected APDD performance of RLNC:

Theorem 5. *Given system parameters K , N , and P_e , the overall APDD performance of RLNC is:*

$$E[D_R] = \frac{1}{1 - P_e} E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right], \quad \{w_n\}_{n=1}^N \sim B(K, P_e) \quad (10)$$

$$\approx \frac{KP_e - P_e + 1}{1 - P_e}, \quad \text{when } N \text{ is sufficiently large} \quad (11)$$

The accuracy of our approximation is again confirmed by the simulation results plotted in Fig. 2.

By comparing the above 3 expected APDD under RLNC with the corresponding lower bounds ((4), (5), (7)), we can easily verify that the ratio between the expected APDD of using RLNC and the lower bound converges to 2 from below with increasing w_n or K . Thus, the expected APDD of using RLNC is at most twice of the minimum expected APDD. We thus have the following theorem:

Theorem 6. *RLNC is an APDD 2-approximation technique when coded transmissions are subject to random packet erasures.*

Since RLNC requires block decoding, any throughput optimal LNC techniques that can provide early packet decodings will outperform RLNC in terms of APDD. Thus, all LNC techniques can approximation APDD with a ratio of at most 2. Next, we will raise an example in which this ratio is $4/3$. These two results yields the following bounds on the approximation ratio of throughput-optimal LNC techniques:

Theorem 7. *The APDD approximation ratio of all throughput-optimal LNC techniques is between $\frac{4}{3}$ and 2.*

Proof. Since approximation ratio is the highest ratio among all settings, to prove that $\beta \geq \frac{4}{3}$, it suffices to show an instance in which the APDD of all throughput-optimal LNC techniques is $\frac{4}{3}$ times of the minimum.

Consider an instance \mathbf{A} of the APDD minimization problem with 2 data packets and N receivers. Receiver r_1 only wants \mathbf{p}_1 , receiver r_2 only wants \mathbf{p}_2 , and all the remaining $N - 2$ receivers want both \mathbf{p}_1 and \mathbf{p}_2 . Further assume that $\{P_{e,n}\}_{n=1}^N = 0$, so that $E[D] = D$. When N is sufficiently large, the APDD is minimized if \mathbf{p}_1 and \mathbf{p}_2 are transmitted separately using two transmissions. This yields $D_{\min} = 1.5$. On the other hand, if throughput-optimal techniques are applied, \mathbf{p}_1 and \mathbf{p}_2 must be combined in the first transmission due to r_1 and r_2 , which does not allow the remaining $N - 2$ receivers to decode in the first transmission. These receivers can only decode in the second transmission. Consequently, the APDD is 2 when N is sufficiently large, which is $\frac{4}{3}$ times of the minimum. \square

B. IDNC

A common feature of IDNC techniques, including strict IDNC (S-IDNC) [12], [16], [30] and general IDNC (G-IDNC) [13], [15], [31], [32], is that every coded packet is the binary XOR of a subset of the data packets. Another common feature of IDNC techniques is “memoryless decoding”: the receivers discard the coded packets that are innovative but not instantly decodable rather than storing them for future decodings². These features reduce decoding complexity, and motivates the sender to send coded packets that are instantly decodable to as many receivers as possible. Such decoding helps reduce APDD, but at a cost of an unbounded degradation of the throughput:

Theorem 8. *IDNC techniques are not throughput-approximation techniques.*

Proof. If IDNC techniques are throughput-approximation techniques, then when there are not packet erasures, the BCT U_I provided by IDNC techniques should be within a constant multiple of the minimum BCT U_{\min} that any LNC techniques can offer. Thus, to prove the theorem, we only need a counter example in which U_I is not within a constant multiple of U_{\min} .

Our counter example involves two types of SFMs:

- $\mathbf{A}_1(K)$: every pair of data packets is wanted by a different receiver. Thus, there are $N = \frac{K(K-1)}{2}$ receivers in total;
- $\mathbf{A}_2(K, m)$: every data packet is wanted by m different receivers. Every pair of data packets is wanted by a different receiver. Thus, there are $N = mK + \frac{K(K-1)}{2}$ receivers in total.

Note that $U_{\min} = 2$ for $\mathbf{A}_1(K)$, which can be achieved by any throughput-optimal LNC techniques.

We prove the theorem by proving that $U_I = \lceil \log_2 K \rceil + 1$ for $\mathbf{A}_1(K)$, where $\lceil x \rceil$ denotes the smallest integer greater than x .

Given $\mathbf{A}_1(K)$, the transmission starts by sending as c_1 the XOR of an arbitrary $m_1 \geq 1$ data packets in $\mathbf{A}_1(K)$. The resulted SFM consists of two sub-SFMs: 1) an $\mathbf{A}_1(m_1)$, which contains the m_1 data packets and the receivers who want 2 data packets from c_1 and thus discard c_1 ; 2) an $\mathbf{A}_2(K - m_1, m_1)$, which contains the remaining $K - m_1$ data packets and the remaining receivers, which either has decoded

²There are recent works in IDNC literature that allow the receiver to store at least one innovative but not instantly decodable packet [19].

one wanted data packet from c_1 and still wants one data packet from $K - m_1$, or wants 2 data packets from $K - m_1$.

These two sub-SFMs are independent in the sense that an IDNC coded packet of $A_1(m_1)$ and an IDNC coded packet of $A_2(K - m_1, m_1)$ can be XOR-ed and sent without affecting their decodability at their targeted receivers.

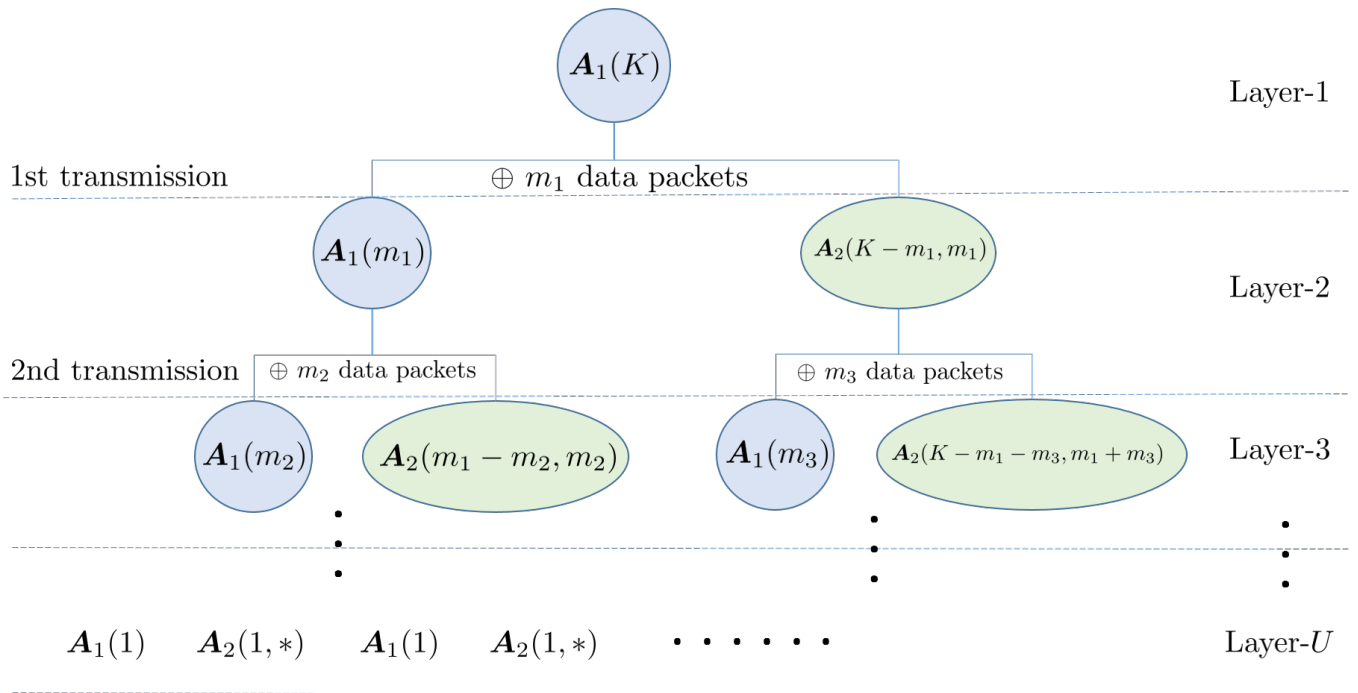
Similarly, we can show that after sending the XOR of any arbitrary m_3 data packets from $A_2(K - m_1, m_1)$, the resulted SFM consists of two independent sub-SFMs: an $A_1(m_3)$ and an $A_2(K - m_1 - m_3, m_1 + m_3)$.

Continuing the logic, after the u -th transmission ($u \geq 1$), $A_1(K)$ is split into 2^{u-1} type-1 sub-SFMs and 2^{u-1} type-2 sub-SFMs. Only sub-SFMs that consists of a single data packet can be completed in one coded transmission and be removed. The evolution of $A_1(K)$ is demonstrated in a layered graph in Fig. 3. The u -th layer corresponds to the SFM before the u -th coded transmission. The total number of coded transmissions is thus the number of layers plus one. It is clear that the minimum number of layers is $\lceil \log_2 K \rceil$, which is achieved by XOR-ing half of the data packets from each sub-SFM. Thus, $U_I = \lceil \log_2 K \rceil + 1$. \square

Our theorem indicates that all IDNC techniques are only heuristics for BCT minimization and throughput maximization. A large BCT can, in turn, hurts APDD, as there are data packets decoded with large delays. In the example raised in the above proof, when IDNC techniques are applied, there are receivers who decode their second wanted data packet after $\lceil \log_2 K \rceil + 1$ transmissions. Even if they decode their first wanted packet after the first transmission, their APDD is still as high as $\lceil \log_2 K \rceil / 2 + 1$, which can be unlimitedly larger than an APDD of 2 offered by RLNC. Therefore, IDNC techniques cannot approximate the APDD of at least a subset of the receivers. However, it is still an open problem whether IDNC techniques can approximate the APDD across all receivers.

V. AN IMPROVED APDD APPROXIMATION LNC TECHNIQUE

In this section, we propose hypergraphic LNC (HLNC), a novel low complexity throughput-optimal and APDD-approximation technique built upon a novel hypergraph model of receivers' packet reception



The $(U + 1)$ -th transmission will finish the broadcast.

Fig. 3: The decoding evolution of $A_1(K)$. (Note that '*'s are arbitrary positive integers whose values depend on the coded packets.)

state. Coded packets generated by HLNC have the following two features:

- 1) every coded packet is innovative to every receiver; and
- 2) every coded packet is able to offer at least one early packet decoding.

The first feature ensures the throughput-optimality and the APDD-approximation of HLNC. The second feature further ensures that both the APDD and decoding complexity of HLNC are lower than RLNC, as RLNC requires block decoding.

A. The Basic Form of HLNC

We introduce our technique by first generalizing the concept of *Wants* set \mathcal{W}_n :

Definition 4. The *Wants set* \mathcal{W}_n of a receiver r_n is the set of data packets not yet decoded by r_n .

The subtle yet important difference between this new definition and the previous one is that the new one explicitly includes in \mathcal{W}_n the received (from coded packets) but undecodable data packets.

We then model the packet reception state $\{\mathcal{W}_n\}$ using the following hypergraph.

Definition 5. In the hypergraph model $\mathcal{H}(\mathcal{V}, \mathcal{E})$ of the receivers' packet reception state, each vertex $v \in \mathcal{V}$ represents a data packet, i.e., $v_k \leftrightarrow \mathbf{p}_k$. Each hyperedge $e \in \mathcal{E}$ represents the Wants set of a receiver, i.e., $e_n \leftrightarrow \mathcal{W}_n$, by connecting the data packets/vertices in \mathcal{W}_n .

An example of \mathcal{H} is plotted in Fig. 1(b), where 6 data packets and 3 receivers are modelled as a hypergraph with 6 vertices and 3 hyperedges.

We further denote by \mathcal{V}_c a minimal vertex cover of \mathcal{H} , and denote the corresponding packet set as \mathcal{M}_c . \mathcal{V}_c is a subset of \mathcal{V} satisfying that 1) it is incident to every hyperedge, i.e., $|e_n \cap \mathcal{V}_c| > 0 \forall n \in [1, N]$, and 2) there is a least one single incidence, i.e. $\exists n : |e_n \cap \mathcal{V}_c| = 1$. Due to these two features, a properly generated (will be discussed soon) coded packet of \mathcal{M}_c has the following two properties:

- 1) It is innovative to every receiver, because every receiver wants at least one data packet from \mathcal{M}_c , i.e., $|\mathcal{W}_n \cap \mathcal{M}_c| > 0 \forall n \in [1, N]$;
- 2) It allows at least one receiver to instantly decodable one wanted data packet, because there is at least one receiver who only wants one data packet from \mathcal{M}_c , i.e., $\exists n : |\mathcal{W}_n \cap \mathcal{M}_c| = 1$.

Therefore, the core of HLNC is to keep updating \mathcal{H} and sending coded packets generated using \mathcal{V}_c . The basic HLNC technique is outlined in Algorithm 1. A complete example of the basic HLNC is given in Example 2 at the end of this section.

Algorithm 1 HLNC Wireless Broadcast

- 1: Input: the initial packet reception state \mathcal{H} ;
 - 2: **while** Not all receivers have decoded all wanted data packets **do**
 - 3: The sender updates \mathcal{H} ;
 - 4: The sender broadcasts a properly generated coded packet c of a minimal vertex cover \mathcal{V}_c of \mathcal{H} ;
 - 5: Every receiver that receives this coded packet tries to decode by solving linear equations(s).
 - 6: **end while**
-

Coded packets must be properly generated to ensure their innovativeness to every receiver. This can be easily accomplished either asymptotically, by random coefficients chosen from a large \mathbb{F}_q (as RLNC),

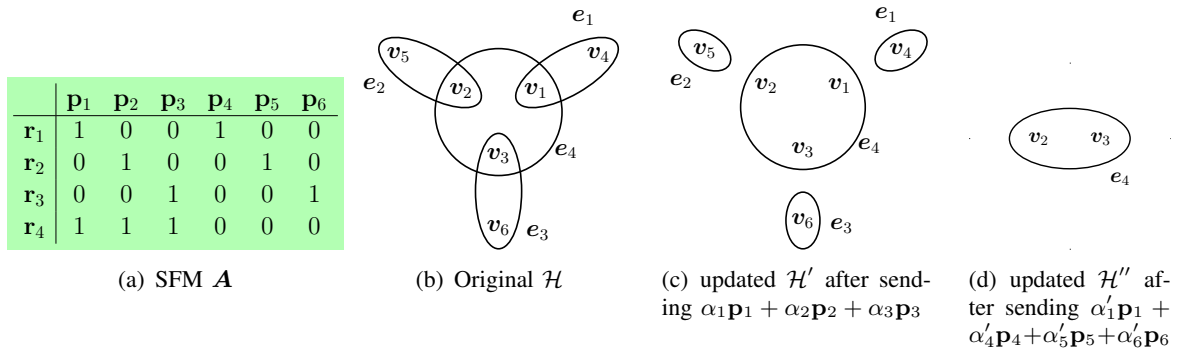


Fig. 4: Hypergraph update

or deterministically, by coefficients generated using deterministic LNC techniques such as [8]³.

On the other hand, a minimal vertex cover can be found in polynomial-time by simple heuristic algorithms. We propose such an algorithm in Algorithm 2, which prioritizes data packets based on their popularity.

One way to understand HLNC is that HLNC aims at achieving throughput optimality by encoding the minimum number of data packets together. Such minimization will minimize the intersection between the coding set and receivers' Wants sets to a minimum of one data packet, which makes such data packets instantly decodable to the corresponding receivers. We note that although [10] has a similar aim, the way how the coding set is determined is completely different, and there is no claim on guaranteed instantly packet decodings nor guaranteed APDD approximation.

Example 2. Consider the SFM in Fig. 4(a), with its hypergraph \mathcal{H} plotted in Fig. 4(b). For simplicity we assume erasure-free transmissions. Then, the transmissions using HLNC is as follows:

- 1) A minimal vertex cover of \mathcal{H} is $\mathcal{V}_c = \{v_1, v_2, v_3\}$. Thus, $\mathbf{X} = \alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \alpha_3 \mathbf{p}_3$ is sent, where $\{\alpha_1, \alpha_2, \alpha_3\}$ are coefficients chosen from $\mathbb{F}_q \setminus \{0\}$. Since receivers r_1, r_2, r_3 only want $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ from \mathbf{X} , respectively, they can decode them from \mathbf{X} . Hence, v_1, v_2, v_3 are removed from e_1, e_2, e_3 , respectively. On the other hand, r_4 cannot decode any data packet. It only holds \mathbf{X} , and thus e_4 is still incident to $\{v_1, v_2, v_3\}$. The updated graph \mathcal{H}' is plotted in Fig. 4(c);
- 2) A minimal vertex cover of \mathcal{H}' is $\mathcal{V}'_c = \{v_1, v_4, v_5, v_6\}$. Thus, $\mathbf{X}' = \alpha'_1 \mathbf{p}_1 + \alpha'_4 \mathbf{p}_4 + \alpha'_5 \mathbf{p}_5 + \alpha'_6 \mathbf{p}_6$ is sent. Since receivers r_1, r_2, r_3, r_4 only want $\mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6, \mathbf{p}_1$ from \mathbf{X}' , respectively, they can

³We note that the coding set selection strategies of these techniques are completely different from HLNC. For example, RLNC selects all the data packets, whilst the technique in [8] selects through solving an NP-hard hitting set problem

Algorithm 2 Find a minimal hypergraph vertex cover

- 1: Input: A hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$, an empty vertex set \mathcal{V}_c ;
 - 2: Weigh each vertex with the number of hyperedges incident to it; (*The weight of a vertex is indeed the number of receivers which want the corresponding data packet.*)
 - 3: **while** there are still vertices in \mathcal{H} **do**
 - 4: Add to \mathcal{V}_c the vertex with the largest weight;
 - 5: Update \mathcal{H} by: 1) removing this vertex from \mathcal{H} ; 2) removing from \mathcal{H} all hyperedges incident to this vertex; 3) removing from \mathcal{H} all vertices that do not have any hyperedge incident to them;
 - 6: **end while**
 - 7: Output \mathcal{V}_c as a minimal hypergraph vertex cover.
-

decode them from \mathbf{X}' . Hence, v_4, v_5, v_6, v_1 are removed from e_1, e_2, e_3, e_4 , respectively. Since receivers $\{r_1, r_2, r_3\}$ are satisfied, $\{e_1, e_2, e_3\}$ are completely removed from \mathcal{H} . On the other hand, r_4 holds a linear equation of $\alpha_2 \mathbf{p}_2 + \alpha_3 \mathbf{p}_3 = \mathbf{X} - \alpha_1 \mathbf{p}_1$, where the value of the RHS is known to r_4 . Since r_4 still wants $\{\mathbf{p}_2, \mathbf{p}_3\}$, e_4 is incident to $\{v_2, v_3\}$. The updated graph \mathcal{H}'' is plotted in Fig. 4(d).

- 3) A minimal vertex cover of \mathcal{H}'' is v_2 . Thus, \mathbf{p}_2 is sent alone, which will allow r_4 to obtain \mathbf{p}_2 , and then decode \mathbf{p}_3 from equation $\alpha_2 \mathbf{p}_2 + \alpha_3 \mathbf{p}_3 = \mathbf{X} - \alpha_1 \mathbf{p}_1$. The coded broadcast will then be completed.

B. Implementation under Different Feedback Availability

The basic HLNC proposed in Algorithm 1 works when the sender can update \mathcal{H} after every transmission. This requires fully-online receiver feedback, namely every receiver sends a feedback every time it receives a coded packet. However, it could be expensive or even impossible to collect feedback after every transmission. For example, in time-division-duplex systems the sender has to stop and listen to the feedback [6], [33]–[35]. Thus, it is desirable to develop implementations of HLNC with semi-online feedback and without intermediate feedback at all (off-line). The challenge here is how to update \mathcal{H} locally without losing throughput-optimality and APDD-approximation.

We solve this challenge by our semi-online HLNC proposed in Algorithm 3. At a high level, we partition the transmissions into semi-online rounds. During each semi-online round (Step 5-9 in Algorithm 3), the sender iteratively broadcasts a coded packet c and then update \mathcal{H} locally by assuming c is received by all receivers. The iteration will be terminated if the reception of c will allow one receiver to fully decode all its wanted data packets. After the termination, the sender will collect receiver feedback to correctly update \mathcal{H} , and then enter the next semi-online round.

Due to this termination condition, the locally updated \mathcal{H} always has the same number of hyperedges/receivers as the actual packet reception state \mathcal{H}' at the receivers with. Moreover, since we update \mathcal{H} by assuming all coded packets are received by all receivers, the vertices removed from \mathcal{H} during the updates are at least as many as \mathcal{H}' . Thus, \mathcal{H} is also a subgraph of \mathcal{H}' . Consequently, \mathcal{H} and \mathcal{H}' have the following easily-proved property:

Property 1. *If \mathcal{H} is a subgraph of \mathcal{H}' with the same number of hyperedges, then any minimal vertex cover \mathcal{V}_c of \mathcal{H} is also a minimal vertex cover of \mathcal{H}' .*

This property immediately indicates that the coded packets generated by semi-online HLNC have the same features as those generated by the fully-online basic HLNC, namely, 1) innovative to every receiver, and 2) always provide early packet decodings. This is because the minimal vertex covers we found in each semi-online round are also minimal vertex covers of the actual \mathcal{H}' at the receivers.

Therefore, semi-online HLNC is throughput-optimal and APDD-approximating as its fully-online counterpart. As we will see in the simulation results presented in the next section, there is no visible performance difference between the two.

Based on semi-online HLNC, we can also develop an offline HLNC by not collecting receiver feedback after the first semi-online round, but using the classic RLNC in the remaining transmissions. As we will see in the simulation results presented in the next section, even off-line HLNC can always provide lower APDD than RLNC.

Moreover, HLNC is very robust to feedback loss. If the feedback from a subset of receivers is lost, the sender can simply update the hypergraph by assuming that these receivers have not received any

previous packets.

Last but not least, HLNC is fully compatible with RLNC and offers seamless switch. The sender can keep sending RLNC packets until one round of receiver feedback is received. Based on the feedback, the sender conducts one HLNC semi-online round, and then switches back to RLNC until the reception of the next round of receiver feedback.

Algorithm 3 HLNC Wireless Broadcast with Semi-online Feedback

- 1: Input: the initial packet reception state \mathcal{H} ;
 - 2: **while** Not all receivers have decoded all wanted data packets **do**
 - 3: The sender updates \mathcal{H} by collecting receiver feedback (waived for the first transmission);
 - 4: The sender broadcasts a properly generated coded packet \mathbf{c} of a \mathcal{V}_c of \mathcal{H} ;
 - 5: **while** No receiver can fully decode all its wanted data packets after receiving \mathbf{c} **do**
 - 6: The sender updates \mathcal{H} locally by assuming \mathbf{c} is received by all receivers;
 - 7: The sender broadcasts a properly generated coded packet \mathbf{c} of a \mathcal{V}_c of \mathcal{H} ;
 - 8: Every receiver that receives this coded packet tries to decode by solving linear equations(s).
 - 9: **end while**
 - 10: **end while**
-

VI. SIMULATION RESULTS

In this section, we numerically compare the APDD performance of the proposed technique with some existing techniques, as well as the lower bound on APDD. In total, there are 6 different APDD we will compare. They are abbreviated and explained as follows:

- 1) “Fully-”: the APDD of fully-online HLNC;
- 2) “Semi-”: the APDD of semi-online HLNC;
- 3) “Off-”: the APDD of offline HLNC;
- 4) “ $E[D]$ ”: the lower bound on the expected APDD of LNC derived in (7);
- 5) “ $E[D_R]$ ”: the expected APDD of RLNC derived in (11);

6) “G-IDNC”: the APDD of a heuristic GIDNC algorithm adapted from [13] when fully-online feedback is collected. We note that there has not been any optimal G-IDNC algorithms in the literature.

In our simulations, there are $K = 15$ data packets, $N \in [5, 100]$ receivers. The packet erasure probabilities are $\{P_{e,n}\}_{n=1}^N = 0.2$. For each value of N , we simulate the broadcast of 10^5 packet blocks, and then make average on their APDD. The simulation results are plotted in Fig. 5, from which we observe that:

- The APDD performance of our technique outperforms the existing techniques. This superiority holds even when our technique is implemented under the off-line scheme;
- The fully- and semi-online schemes share the same performance. This result matches our expectation. Their performance is better than the off-line one;
- The APDD of RLNC is always within a constant factor of the lower bound, indicating that RLNC is an approximation technique, and so is our technique. On the other hand, the APDD of the heuristic G-IDNC is unbounded, indicating that it is not an approximation technique. We further note that one should not read the approximation ratio from the figure, because the approximation ratio is achieved when K is sufficiently large, as have discussed in Section IV before Theorem 6.

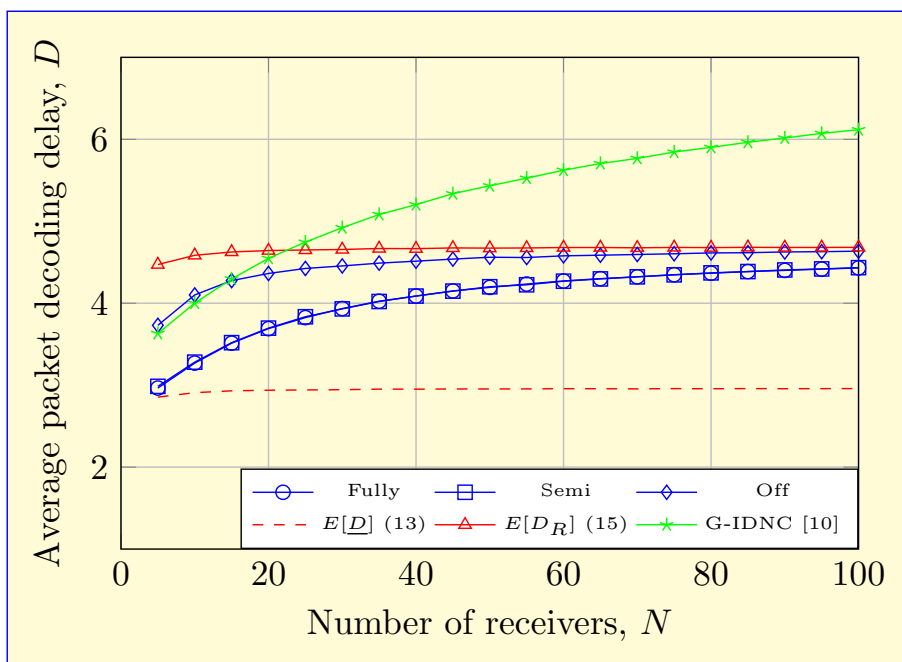


Fig. 5: The APDD performance of different NC techniques.

In addition, we compare the amount of feedback collected under fully- and semi-online HLNC. The results are plotted in Fig. 6. We observe that semi-online HLNC can reduce up to 30% feedback from the fully-online one when the number of receivers is small. The reduction becomes marginal with increasing number of receivers because the probability of having a receiver who only wants one data packet after a certain semi-online round increases. When this happens, semi-online HLNC has to collect feedback after only one transmission according to Algorithm 3, which makes it equivalent to fully-online HLNC.

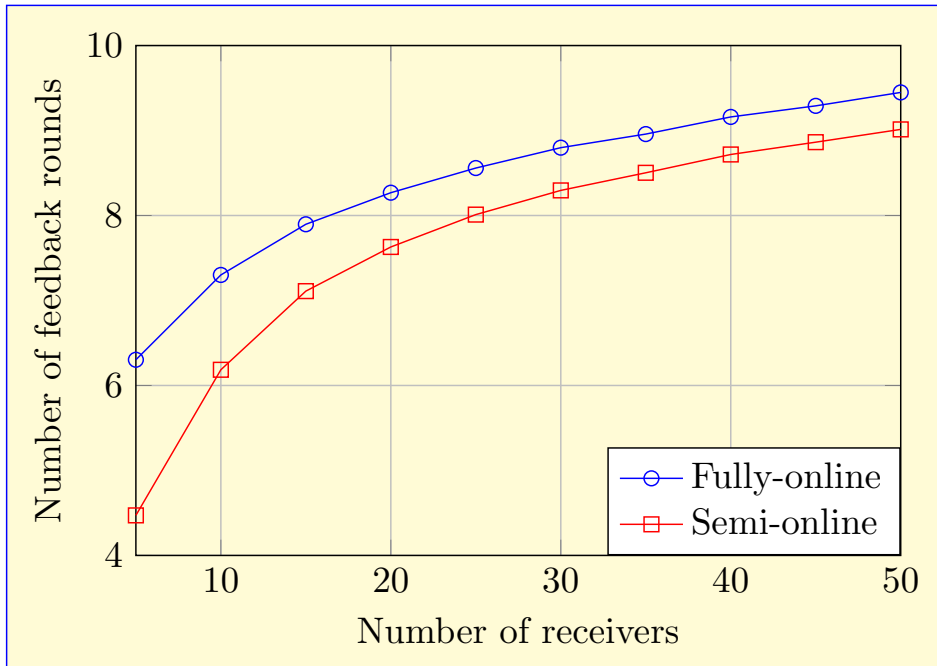


Fig. 6: The Amount of feedback collected under the fully- and semi-online schemes.

VII. CONCLUSION

In this paper, we conducted a comprehensive study on the APDD performance of LNC techniques in wireless broadcast with packet erasures. By deriving lower bounds on the expected APDD of LNC techniques using a conceptual perfect LNC technique, we showed that the APDD of efficient LNC techniques should not scale up with increasing number of receivers. Then by proving the NP-hardness of achieving these lower bounds, we proved the NP-hardness of the APDD minimization problem.

Although the optimal APDD is intractable to achieve, we proved that it can be approximated with a ratio between $4/3$ and 2 by throughput-optimal LNC techniques. Therefore, throughput and APDD can be jointly improved rather than trading each other off. However, such joint improvement does not

necessarily hold for every LNC technique. For example, we proved that all IDNC techniques cannot approximate the optimal throughput, and such sub-optimality in turn makes IDNC techniques unable to guarantee an APDD approximation ratio for at least a subset of receivers.

We then proposed HLNC, a novel hypergraph-based LNC technique that combines all the advantages of RLNC and IDNC: It is throughput-optimal, APDD-approximating, computational friendly, and it always provides instant packet decodings. Moreover, its performance does not degrade even if receiver feedback is not collected after every transmission. Our extensive simulations showed that the APDD of HLNC outperforms RLNC and a heuristic general IDNC technique under all tested system parameter settings.

In the future, we are interested in extending our hypergraph model and APDD analysis to other network coding and index coding problems. We are also interested in applying our technique to other network models such as cooperative data exchange [36], [37].

APPENDIX A

PROOF OF LEMMA 1

We prove that it is NP-complete to determine whether an r -uniform hypergraph is size- r strong colorable or not, for any $r \geq 3$. Our method is a reduction from the k -coloring problem of graphs.

Given an arbitrary graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. For every edge $e_n = (v_i, v_j)$ we construct a hyperedge e'_n by adding $r - 2$ dummy vertices $v^{n,1} \dots v^{n,r-2}$ to e_n . The result is an r -uniform hypergraph $\mathcal{H}(\mathcal{V}', \mathcal{E}')$ that has $|\mathcal{V}'| = |\mathcal{V}| + |\mathcal{E}| \cdot (r - 2)$ vertices. If \mathcal{G} can be colored using r colors, then in any hyperedge $e'_n = (v_i, v_j, v^{n,1} \dots v^{n,r-2})$, v_i and v_j are colored differently using 2 colors. By assigning the remaining $r - 2$ colors to the $r - 2$ dummy vertices in e'_n greedily, all vertices in e'_n are colored differently. We thus obtain a size- r strong coloring of \mathcal{H} . On the other hand, if \mathcal{H} can be strong colored using r colors, then by removing all the dummy vertices, we obtain an r -coloring of \mathcal{G} . It is well known that it is NP-complete to determine whether a graph is r colorable or not, for any $r \geq 3$. Hence, it is NP-complete to determine whether an r -uniform hypergraph is size- r strong colorable. ■

APPENDIX B

PROOF OF THEOREM 1

We first assume that r_n is satisfied (i.e., obtains all data packets) after U_n coded transmissions. Hence the packet reception pattern \mathbf{u}_n of r_n takes a form of $[u_1, \dots, u_{w_n-1}, U_n]$. Let $\mathbf{u}' = [u_1, \dots, u_{w_n-1}]$, then the APDD under a given \mathbf{u} is $(\|\mathbf{u}'\| + U_n)/w_n$. It is intuitive that all the possible \mathbf{u}' happens with the same probability. Denote the set of all possible \mathbf{u}' by \mathcal{U} , then the expected lower bound under given \mathcal{W}_n and U_n , denoted by $E[\underline{D}_n|(U_n, w_n)]$, is calculated as:

$$\begin{aligned} E[\underline{D}_n|(U_n, w_n)] &= \frac{1}{|\mathcal{U}|} \sum_{\mathbf{u}' \in \mathcal{U}} \frac{\|\mathbf{u}'\| + U_n}{w_n} \\ &= \frac{U_n}{w_n} + \frac{1}{|\mathcal{U}|} \sum_{\mathbf{u}' \in \mathcal{U}} \frac{\|\mathbf{u}'\|}{w_n} \end{aligned} \quad (12)$$

We now show that all \mathbf{u}' are symmetric. Given $\mathbf{u}' = [u_1, \dots, u_{w_n-1}]$, by letting $u''_i = U_n - u_i$, the resultant $\mathbf{u}'' = [u''_1, \dots, u''_{w_n-1}]$ is the mirror of \mathbf{u}' against $U_n/2$. Obviously, \mathbf{u}'' also belongs to \mathcal{U} , and it holds that $\|\mathbf{u}'\| + \|\mathbf{u}''\| = (w_n - 1)U_n$. Hence, there are $|\mathcal{U}|/2$ such pairs, and thus the above equation can be simplified to:

$$E[\underline{D}_n|(U_n, w_n)] = \frac{U_n}{w_n} + \frac{1}{2} \frac{(w_n - 1)U_n}{w_n} = \frac{U_n}{2} + \frac{U_n}{2w_n} \quad (13)$$

Then, by noting that the expected number of coded transmissions for a receiver to be satisfied is $E[U_n|w_n] = w_n/(1 - P_e)$, we have:

$$E[\underline{D}_n|w_n] = \frac{E[U_n|w_n]}{2} + \frac{E[U_n|w_n]}{2w_n} = \frac{w_n + 1}{2(1 - P_e)} \quad (14)$$

which proves Theorem 1. ■

APPENDIX C

PROOF OF THEOREM 2

Here we only need to prove that $E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] \approx KP_e - P_e + 1$ when N is sufficiently large. We first expand $E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right]$ into its series form:

$$E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] = E \left[\frac{w_1^2}{\sum_{n=1}^N w_n} \right] + E \left[\frac{w_2^2}{\sum_{n=1}^N w_n} \right] + \cdots + E \left[\frac{w_N^2}{\sum_{n=1}^N w_n} \right] \quad (15)$$

Since $\{w_n\}_{n=1}^N$ are i.i.d. distributed, the N addends in the above equation have the same value. Thus,

$$E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] = N \cdot E \left[\frac{w_1^2}{\sum_{n=1}^N w_n} \right] \quad (16)$$

$$= N \cdot E \left[\frac{w_1^2}{w_1 + \sum_{n=2}^N w_n} \right] \quad (17)$$

Then according to the law of larger numbers, the value of $\sum_{n=2}^N w_n$ will approach to its mean $(N-1)KP_e$ when N is sufficiently large. Thus,

$$E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] \approx N \cdot E \left[\frac{w_1^2}{w_1 + (N-1)KP_e} \right], \quad \text{when } N \text{ is sufficiently large} \quad (18)$$

$$= E \left[\frac{w_1^2}{KP_e + \frac{w_1 - KP_e}{N}} \right] \quad (19)$$

$$\approx E \left[\frac{w_1^2}{KP_e} \right], \quad \text{when } N \text{ is sufficiently large} \quad (20)$$

Then, since $w_1 \sim B(K, P_e)$, we have $E[w_1] = KP_e$ and $Var[w_1] = KP_e - KP_e^2$. Hence, we have $E[w_1^2] = E[w_1]^2 + Var[w_1] = K^2P_e^2 + KP_e - KP_e^2$, and thus $E \left[\frac{\sum_{n=1}^N w_n^2}{\sum_{n=1}^N w_n} \right] \approx KP_e - P_e + 1$. ■

REFERENCES

- [1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [2] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE transactions on information theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782–795, 2003.
- [4] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [5] M. Nistor, D. E. Lucani, T. T. V. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1084–1093, May 2011.

- [6] D. E. Lucani, M. Médard, and M. Stojanovic, "Random linear network coding for time-division duplexing: Field size consideration," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2009.
- [7] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, "Network coding for mobile devices - systematic binary random rateless codes," in *Proc. IEEE Int. Conf. Communications (ICC) workshop*, 2009.
- [8] H. Y. Kwan, K. W. Shum, and C. W. Sung, "Generation of innovative and sparse encoding vectors for broadcast systems with feedback," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2011, pp. 1161–1165.
- [9] M. Yu, P. Sadeghi, and N. Aboutorab, "On deterministic linear network coded broadcast and its relation to matroid theory," in *IEEE Information Theory Workshop (ITW)*, 2014, pp. 536–540.
- [10] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. of NetCod*, 2008.
- [11] M. Yu, A. Sprintson, and P. Sadeghi, "On minimizing the average packet decoding delay in wireless network coded broadcast," in *Proc. IEEE Int. Symp. Network Coding (NetCod)*, 2015, pp. 1–5.
- [12] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER: Efficient retransmission scheme for wireless LANs," in *Proc. ACM CoNEXT*, 2007.
- [13] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. IEEE Int. Conf. Communications (ICC)*, May 2010, pp. 1–5.
- [14] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. on Wireless Commun. and Netw.*, pp. 1–14, Jan. 2010.
- [15] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Trans. Networking*, vol. 23, no. 5, pp. 1553–1567, 2015.
- [16] M. Yu, P. Sadeghi, and N. Aboutorab, "Performance characterization and transmission schemes for instantly decodable network coding in wireless broadcast," *European Journal on Advances in Signal Processing (EURASIP)*, vol. 2015, no. 1, p. 94, 2015.
- [17] R. Costa, D. Munaretto, J. Widmer, and J. Barros, "Informed network coding for minimum decoding delay," in *Proc. IEEE Int. Conf. on Mobile Ad Hoc and Sensor System, (MASS)*, 2008, pp. 80–91.
- [18] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, 2011.
- [19] N. Aboutorab, S. Sorour, and P. Sadeghi, "O2-GIDNC: Beyond instantly decodable network coding," in *Proc. IEEE Int. Symp. Network Coding (NetCod)*, 2013, pp. 1–6.
- [20] N. Aboutorab, P. Sadeghi, and S. Sorour, "Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1296–1309, Mar. 2014.
- [21] S. Sorour, N. Aboutorab, T. Y. Al-Naffouri, and M.-S. Alouini, "A graph model for opportunistic network coding," in *Proc. IEEE Symp. Network Coding (NetCod)*, 2015, pp. 26–30.
- [22] M. Yu, N. Aboutorab, and P. Sadeghi, "Rapprochement between instantly decodable and random linear network coding," in *Proc. IEEE Symp. Information Theory (ISIT)*, 2013, pp. 3090–3094.
- [23] M. Yu, P. Sadeghi, and N. Aboutorab, "From instantly decodable to random linear network coding," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3943–3955, Oct. 2014.

- [24] M. Yu, P. Sadeghi, and A. Sprintson, "Feedback-assisted random linear network coding in wireless broadcast," in *Proc. IEEE Global Communications Conference (GLOBECOM) Workshop*, 2016, pp. 1–6.
- [25] S. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Proc. IEEE Information Theory Workshop (ITW)*, Lake Tahoe, NV, 2007.
- [26] M. A. R. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *Proc. IEEE INFOCOM Workshops*, 2008, pp. 1–4.
- [27] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.
- [28] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Mar. 2011.
- [29] S. Brahma and C. Fragouli, "Pliable index coding: The multiple requests case," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2013, pp. 1142–1146.
- [30] J. K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. 5th Workshop on Network Coding, Theory, and Applications (NetCod)*, 2009.
- [31] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE Journ. Sel. Areas in Comm.*, vol. 29, no. 5, pp. 1094–1105, 2011.
- [32] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc. IEEE Symp. Network Coding (NetCod)*, 2013, pp. 1–6.
- [33] D. E. Lucani, M. Muriel, and M. Stojanovic, "Broadcasting in time-division duplexing: A random linear network coding approach," in *Proc. 5th Workshop on Network Coding, Theory, and Applications (NetCod)*, 2009.
- [34] D. Lucani, M. Médard, and M. Stojanovic, "Systematic network coding for time-division duplexing," in *proc. IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jun. 2010, pp. 2403–2407.
- [35] ———, "On coding for delay – Network coding for time division duplexing," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2330–2348, Apr. 2012.
- [36] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *IEEE Information Theory Workshop (ITW)*, 2010, pp. 1–5.
- [37] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *IEEE Int. Symp. Information Theory (ISIT)*, 2010, pp. 1888–1892.