Yu *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:94
DOI 10.1186/s13634-015-0279-x

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Performance characterization and transmission schemes for instantly decodable network coding in wireless broadcast

Mingchao Yu*, Parastoo Sadeghi and Neda Aboutorab

Abstract

We consider broadcasting a block of packets to multiple wireless receivers under random packet erasures using instantly decodable network coding (IDNC). The sender first broadcasts each packet uncoded once, then generates coded packets according to receivers' feedback about their missing packets. We focus on strict IDNC (S-IDNC), where each coded packet includes at most one missing packet of every receiver. But, we will also study its relation with generalized IDNC (G-IDNC), where this condition is relaxed. We characterize two fundamental performance limits of S-IDNC: (1) the number of transmissions to complete the broadcast, which measures throughput and (2) average packet decoding delay, which measures how fast each packet is decoded at each receiver on average. We derive a closed-form expression for the expected minimum number of transmissions in terms of the number of packets and receivers and the erasure probability. We prove that it is NP-hard to minimize the average packet decoding delay of S-IDNC. We also prove that the graph models of S- and G-IDNC share the same chromatic number. Next, we design efficient S-IDNC transmission schemes and coding algorithms with full/intermittent receiver feedback. We present simulation results to corroborate the developed theory and compare our schemes with existing ones.

Keywords: Wireless broadcast, Network coding, Throughput, Packet decoding delay, Feedback

1 Introduction

The broadcast nature of wireless medium allows one sender to simultaneously serve multiple receivers who are interested in the same data. We consider a block-based wireless broadcast system where a sender wishes to deliver a block of data packets to a set of receivers. The channels between the sender and the receivers are subject to independent random packet erasures. In such systems, a traditional approach is to retransmit the data packets under a receiver feedback mechanism, such as Automatic-Repeat-reQuest (ARQ) [1]. This approach, though simple, is inefficient in terms of throughput, as the transmitted packets are non-innovative to the receivers who have already received them.

The advent of network coding (NC) [2] starts a new era for high-throughput network coded wireless communications [3–17]. By linearly adding all data packets together with randomly chosen coefficients from a sufficiently large finite field, random linear network coding (RLNC) can almost surely achieve the minimum block completion time [9–11, 18], which is defined as the number of transmissions it takes to complete the broadcast. Due to the inverse relation between block completion time and throughput under a fixed block size, RLNC can almost surely achieve optimal throughput.

However, with RLNC, data packets are block-decoded by solving a set of linear equations, which can only take place after a sufficient number of coded packets have been received. RLNC thus may suffer from high decoding computational load [11], as well as large average packet decoding delay (APDD) [16], which reflects how fast each data packet is decoded at each receiver on average. High decoding computational load may not be affordable by receivers with limited computational and energy resources, such as mobile and sensor receivers [11]. Large APDD is undesirable in applications where individual data packets are useful, such as image transmissions and video streaming [19–21].

*Correspondence: ming.yu@anu.edu.au
Research School of Engineering, Australian National University, Canberra, Australia

To alleviate decoding computational load and APDD, the sender can transmit all the data packets uncoded once at the beginning of the broadcast. This method allows the receivers to directly obtain a subset of data packets. It is also throughput optimal and constitutes the *systematic transmission phase* of RLNC [11]. But in the subsequent *coded transmission phase*, RLNC still requires block decoding and suffers from high decoding computational load and APDD.

To further mitigate these issues in the coded transmission phase, instantly decodable network coding (IDNC) techniques [3, 12–15] have been introduced. They make online coding decisions based on receivers' feedback about their packet reception state, under the restriction that coding/decoding is over the binary field. In other words, IDNC techniques shift the computational load to the sender, and allow receivers to perform simple binary XOR based instant packet decodings, as the sender (such as a base station) usually has much more computational and energy resources than the receivers.

For a handy example of IDNC techniques, consider the packet reception state in Table 1. There are 4 data packets, $\mathbf{p}_{1,2,3,4}$, and 2 receivers, $R_{1,2}$. R_1 has received $\mathbf{p}_{3,4}$ and only wants $\mathbf{p}_{1,2}$, while R_2 has received $\mathbf{p}_{1,2}$ and only wants $\mathbf{p}_{3,4}$. Consider two IDNC coded packets $\boldsymbol{X}_1 = \mathbf{p}_1 \oplus \mathbf{p}_3$ and $\boldsymbol{X}_2 = \mathbf{p}_2 \oplus \mathbf{p}_4$, where $\oplus$ denotes the binary XOR operator. By transmitting $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and assuming no packet erasures, both receivers can instantly decode one wanted data packet after each transmission. Hence, there are two data packets decoded in the first transmission and two data packets decoded in the second transmission. The corresponding APDD is $\frac{1+1+2+2}{4} = 1.5$. In contrast, if RLNC is applied, all data packets are decoded after the second transmission, yielding an APDD of 2.

We also note from the above example that an IDNC coded packet of $\boldsymbol{X} = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \mathbf{p}_3$ is not instantly decodable to R_1. Restrictions on such packets separate IDNC techniques into two variations. The first one, called strict IDNC (S-IDNC) [13–15], prohibits the transmissions of non-instantly decodable packets to any receiver. Effectively, each coded packet can include at most one wanted data packet of every receiver. The second one, called generalized IDNC (G-IDNC), removes this restriction for more coding opportunities.

Therefore, S-IDNC can be thought of as a sub-class of G-IDNC, in the sense that every valid S-IDNC coded packet is also a valid G-IDNC coded packet. Although G-IDNC has been extensively studied under various wireless broadcast settings, including basic ones [12, 12, 22–28] and those with limited/lossy feedback [29, 30] or with hard deadline [31], most developed algorithms are heuristics, leaving the optimal G-IDNC in terms of throughput and APDD still unknown or intractable due to prohibitively large computational complexity. Hence in this paper, we take a step back, aiming to understand the performance limits and optimal implementations of a sub-class of G-IDNC, namely, S-IDNC. This will facilitate the applications of S-IDNC, while also providing new insights into the more general G-IDNC class.

Table 1 A example of packet reception state

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$
R_1	wants	wants	has	has
R_2	has	has	wants	wants

So far, studies on theoretical performance characterization and implementations of S-IDNC have been quite limited in both breath and depth. S-IDNC was graphically modeled in [13], which then proved that the minimum clique partition solution [13] of the associated graph can be an S-IDNC solution that minimizes the block completion time. However, this solution does not take into account the issues of decoding delay and the robustness of coded transmissions to erasures. S-IDNC has shown to be asymptotically throughput optimal when there are up to three receivers or when the number of data packets approaches infinity [20], but the general relation between the throughput of S-IDNC and system parameters has not been characterized before. In addition and to the best of our knowledge, the minimum packet decoding delay of S-IDNC is still unknown. Moreover, there have not been S-IDNC transmission schemes that can work with intermittent receiver feedback. Another unaddressed problem is a systematic performance comparison between S-IDNC and G-IDNC.

In this paper, we study the above problems and provide the following contributions:

1. We characterize the throughput performance limits of S-IDNC. Specifically, we derive a closed-form expression for the expected minimum block completion time in terms of the number of packets and receivers and their erasure probabilities.
2. We prove that it is NP-hard to minimize the APDD of S-IDNC. We derive an upper bound on the minimum packet decoding delay in terms of the minimum block completion time.
3. We introduce the concept of packet multiplicity, which measures the robustness of data packets against packet erasures. We develop optimal and heuristic algorithms that find S-IDNC solutions with the minimum number of coded packets and with high packet multiplicities. These solutions are better than the minimum clique partition solution identified in [13]. We also design S-IDNC transmission schemes under full and intermittent receiver feedback.

4. We also provide new results on the relation between S-IDNC and G-IDNC. For example, we prove the equivalence between the chromatic number of S- and G-IDNC graphs.

2 System model and notations

2.1 Transmission setup

We consider a block-based wireless broadcast scenario, in which the sender needs to deliver a block of K data packets, denoted by $\mathcal{P}_K = \{\mathbf{p}_k\}_{k=1}^K$, to N receivers, denoted by $\mathcal{R}_N = \{R_n\}_{n=1}^N$ through wireless channels that are subject to independent random packet erasures.

Initially, the K data packets are transmitted uncoded once using K time slots, constituting a *systematic transmission phase* [11]. Then, each receiver provides feedback to the sender about the packets it has received.[1] The complete packet reception state is represented by an $N \times K$ state feedback matrix (SFM) $\boldsymbol{A}$, where $a_{n,k} = 0$ if R_n has already received $\mathbf{p}_k$, and $a_{n,k} = 1$ if R_n has missed (and thus still wants) $\mathbf{p}_k$. The set of data packets wanted by R_n is called the *Wants* set of R_n and is denoted by $\mathcal{W}_n$. The set of receivers who want $\mathbf{p}_k$ is called the *Target* set of $\mathbf{p}_k$ and is denoted by $\mathcal{T}_k$. The size of $\mathcal{T}_k$ is denoted by T_k. Packets with larger T_k are more desired by receivers.

Example 1. *Consider the SFM in Fig. 1a with $K = 6$ data packets and $N = 5$ receivers. The Wants set of R_1 is $\mathcal{W}_1 = \{\mathbf{p}_1, \mathbf{p}_5, \mathbf{p}_6\}$. The Target set of $\mathbf{p}_3$ is $\mathcal{T}_3 = \{R_3, R_5\}$ and thus $T_3 = 2$.*

Then based on $\boldsymbol{A}$, the sender starts the second phase, called the *coded transmission phase*, in which coded packets are transmitted until the broadcast is completed, i.e, until all receivers have recovered all the K data packets. A sketch of the two-phase transmission scheme is plotted in Fig. 2.

2.2 Coded transmission phase: two types of IDNC

In the coded transmission phase, the sender generates IDNC coded packets under the binary field $\mathbb{F}_2$. Explicitly, IDNC coded packets are of the form $\boldsymbol{X} = \bigoplus_{\mathbf{p}_k \in \mathcal{M}} \mathbf{p}_k$, where $\mathcal{M}$ is a selected subset of $\mathcal{P}_K$, and is called an IDNC coding set. There are three possible types of decodability of $\boldsymbol{X}$ at each receiver:

Definition 1.1. *An IDNC coded packet $\boldsymbol{X}$ is instantly decodable for receiver R_n if $\mathcal{M}$ contains exactly one data packet from the Wants set $\mathcal{W}_n$ of R_n, i.e., if $|\mathcal{M} \cap \mathcal{W}_n| = 1$.*

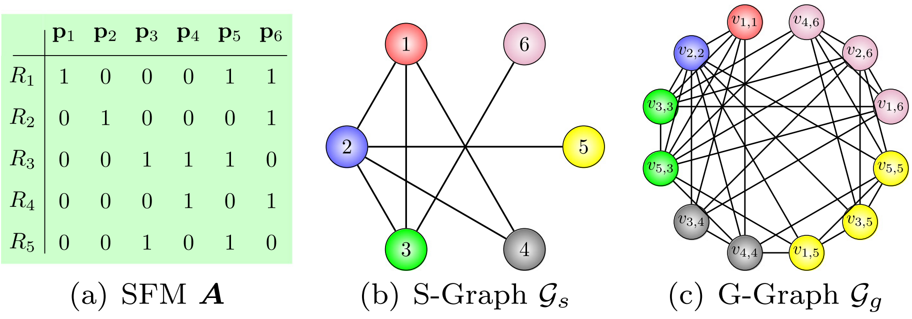
Definition 1.2. *An IDNC coded packet $\boldsymbol{X}$ is non-instantly decodable for receiver R_n if $\mathcal{M}$ contains two or more data packets from the Wants set $\mathcal{W}_n$ of R_n, i.e., if $|\mathcal{M} \cap \mathcal{W}_n| > 1$.*

Definition 1.3. *An IDNC coded packet $\boldsymbol{X}$ is non-innovative for receiver R_n if $\mathcal{M}$ contains no data packets from the Wants set $\mathcal{W}_n$ of R_n, i.e., if $|\mathcal{M} \cap \mathcal{W}_n| = 0$. Otherwise, it is innovative.*

Restrictions on the above three types of packet decodability separate IDNC into two variations. The first one is called strict IDNC (S-IDNC), which is the main subject of our study. It prohibits the transmission of any non-instantly decodable coded packets to any receiver. This restriction implies that any two data packets wanted by the same receiver cannot be coded together. We thus have the concept of conflicting and non-conflicting data packets:

Definition 2. *Two data packets $\mathbf{p}_i$ and $\mathbf{p}_j$ conflict if at least one receiver wants both of them, i.e., if $\exists n : \{\mathbf{p}_i, \mathbf{p}_j\} \subseteq \mathcal{W}_n$. Otherwise they do no conflict.*

An S-IDNC coding set is thus a set of pairwise non-conflicting data packets. The conflicting state between all data packets can be represented by an undirected graph $\mathcal{G}_s(\mathcal{V}, \mathcal{E})$. Each vertex $\boldsymbol{v}_i \in \mathcal{V}$ represents a data packet $\mathbf{p}_i$. Two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are connected by an edge $\boldsymbol{e}_{i,j} \in \mathcal{E}$ if $\mathbf{p}_i$ and $\mathbf{p}_j$ do not conflict. Thus, every complete subgraph of $\mathcal{G}_s$, a.k.a., a clique, represents an S-IDNC coding set. In the rest of the paper, we will use the terms "coded packet",



	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$
R_1	1	0	0	0	1	1
R_2	0	1	0	0	0	1
R_3	0	0	1	1	1	0
R_4	0	0	0	1	0	1
R_5	0	0	1	0	1	0

(a) SFM $\boldsymbol{A}$ (b) S-Graph $\mathcal{G}_s$ (c) G-Graph $\mathcal{G}_g$

Fig. 1 An example of SFM and its S- and G-IDNC graphs

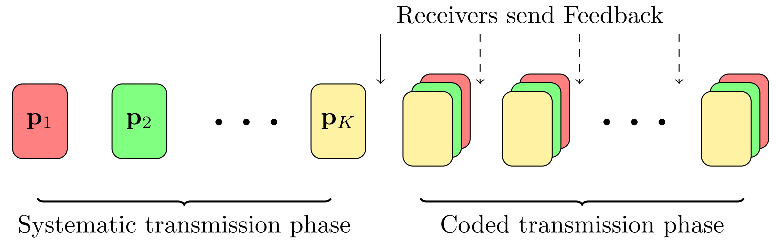


Fig. 2 The considered two-phase transmission schemes. Receiver feedback must be collected by the end of the systematic transmission phase (*solid arrow*). Intermediate feedback during the coded transmission phase (*dashed arrows*) are optional

"coding set", and "clique" interchangeably, and denote the last two by $\mathcal{M}$.

The main limitation of S-IDNC is that a coded packet which is instantly decodable for a large subset of receivers may be prohibited because it is non-instantly decodable for a small subset of receivers. In the second type of IDNC, called generalized IDNC (G-IDNC), the restriction on non-instantly decodable packets is removed for more coding opportunities.[2]

G-IDNC can also be graphically modeled [23]. The difference is that, in the G-IDNC graph $\mathcal{G}_g(\mathcal{V}, \mathcal{E})$, a data packet $\mathbf{p}_k$ wanted by different receivers are individually represented by different vertices $\boldsymbol{v}_{n,k}$, for all $a_{n,k} = 1$. Consequently, the number of vertices in $\mathcal{G}_g$ is equal to the number of "1"s in $\boldsymbol{A}$. Two vertices $\boldsymbol{v}_{m,i}$ and $\boldsymbol{v}_{n,j}$ are connected by an edge if: (1) $i = j$, or (2) if $\mathbf{p}_i \notin \mathcal{W}_n$ and $\mathbf{p}_j \notin \mathcal{W}_m$. In the first case, $\mathbf{p}_i = \mathbf{p}_j$, and thus by sending $\mathbf{p}_i$ both R_m and R_n can decode. In the second case, by sending $\mathbf{p}_i \oplus \mathbf{p}_j$, R_m and R_n can decode $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively, because they already have $\mathbf{p}_j$ and $\mathbf{p}_i$, respectively. Similar to S-IDNC, every clique of $\mathcal{G}_g$ represents a G-IDNC coding set.

We note that an S-IDNC coded packet is always a G-IDNC coded packet, but the reverse is not necessarily true. Below is an example of S- and G-IDNC coded packets.

Example 2. *Consider the SFM and its S- and G-IDNC graphs in Fig. 1. The G-IDNC graph indicates that* $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{5,3}, \boldsymbol{v}_{4,4})$ *is a clique. The corresponding G-IDNC coding set is* $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4)$*, and thus* $\boldsymbol{X}_g = \mathbf{p}_1 \oplus \mathbf{p}_3 \oplus \mathbf{p}_4$ *is a G-IDNC coded packet.* $\boldsymbol{X}_g$ *is instantly decodable for* R_1, R_4, R_5 *because they only want one data packet from* $\boldsymbol{X}_g$. $\boldsymbol{X}_g$ *is non-instantly decodable for* R_3 *because* R_3 *wants both* $\mathbf{p}_3$ *and* $\mathbf{p}_4$. $\boldsymbol{X}_g$ *is non-innovative for* R_2.

Due to the existence of R_3, $\boldsymbol{X}_g$ *is not an S-IDNC coded packet. Whereas the S-IDNC graph indicates that* $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ *is a clique. The corresponding coding set is* $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$*, and thus* $\boldsymbol{X}_s = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \mathbf{p}_3$ *is an S-IDNC coded packet, which can be verified to also correspond to clique* $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{3,3}, \boldsymbol{v}_{5,3})$ *in the G-IDNC graph.*

We then introduce the notion of *IDNC solution*. A set of IDNC coding sets is called an IDNC solution if, upon the reception of the coded packets of all these coding sets, every receiver can decode all its wanted data packets. An S-IDNC solution is denoted by $\mathcal{S}_s$. The set of all S-IDNC solutions of a given SFM is denoted by $\mathbb{S}_s$. Similarly, we can also define $\mathcal{S}_g$ and $\mathbb{S}_g$ for G-IDNC.

For the SFM in Fig. 1, by partitioning the S-IDNC graph into three disjoint cliques, we can obtain, among others, an S-IDNC solution with three cliques/coding sets: $\mathcal{S}_s = \{(\mathbf{p}_1, \mathbf{p}_4), (\mathbf{p}_2, \mathbf{p}_5), (\mathbf{p}_3, \mathbf{p}_6)\}$. We can also partition the S-IDNC graph into four disjoint cliques and obtain $\mathcal{S}_s = \{(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3), \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6\}$. Similarly, a disjoint clique partition of the G-IDNC graph is $\{(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{5,3}, \boldsymbol{v}_{4,4}),\ (\boldsymbol{v}_{3,3}, \boldsymbol{v}_{1,6}, \boldsymbol{v}_{2,6}, \boldsymbol{v}_{4,6}),\ (\boldsymbol{v}_{1,5}, \boldsymbol{v}_{3,5}, \boldsymbol{v}_{5,5}), (\boldsymbol{v}_{3,4}, \boldsymbol{v}_{4,4})\}$, indicating a G-IDNC solution of $\mathcal{S}_g = \{(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4), (\mathbf{p}_3, \mathbf{p}_6), (\mathbf{p}_5), (\mathbf{p}_4)\}$.

To assess the performance of IDNC solutions, we now introduce our measures of throughput and decoding delay.

2.3 Throughput and decoding delay measures

An S-IDNC solution $\mathcal{S}_s$ requires a minimum of $|\mathcal{S}_s|$ coded transmissions. We call $U_{\mathcal{S}_s} \triangleq |\mathcal{S}_s|$ the *minimum block completion time* of $\mathcal{S}_s$. It measures the best throughput of $\mathcal{S}_s$, because the total number of transmissions in the systematic and coded transmission phases is lower bounded by $K + U_{§_s}$, yielding a throughput of $\frac{K}{K+U_{§_s}}$ packet per transmission. We further denote by U_s the absolute minimum block completion time over all the S-IDNC solutions of $\boldsymbol{A}$, i.e., $U_s \triangleq \min\{U_{§_s} : §_s \in \mathbb{S}_s\}$. Similarly, we denote by U_g the absolute minimum block completion time over all the G-IDNC solutions of $\boldsymbol{A}$.

Decoding delay reflects how fast each data packet is decoded by each receiver on average. In this paper, we measure decoding delay by *average packet decoding delay*

(APDD) D, which is the average time it takes for a receiver to decode a data packet. For example, the APDD of all receivers is calculated as:

$$D = \frac{1}{T} \sum_{\forall a_{n,k}=1} u_{n,k}, \tag{1}$$

where $u_{n,k}$ is the time index when R_n decodes $\mathbf{p}_k$, and $T = \sum_{k=1}^{K} T_k$, which is also the number of "1"s in $\boldsymbol{A}$.

Given an IDNC solution $\mathcal{S}$, by letting $u_{n,k}$ be the *first* time index when $\mathcal{S}$ allows R_n to decode $\mathbf{p}_k$, (1) produces the minimum APDD of $\mathcal{S}$. We further denote by D_s (resp. D_g) the absolute minimum APDD over all S- (resp. G-) IDNC solutions of $\boldsymbol{A}$. We also note that in the specific case of an S-IDNC solution $\mathcal{S}_s$, $u_{n,k}$ is indeed the index of the first coding set in $\mathcal{S}_s$ that contains $\mathbf{p}_k$, as every receiver who wants $\mathbf{p}_k$ can decode it from this coding set.

Example 3. *Consider the SFM in Fig. 1a. Suppose that an S-IDNC solution with four coded packets* $\boldsymbol{X}_1 = \mathbf{p}_1 \oplus \mathbf{p}_2$, $\boldsymbol{X}_2 = \mathbf{p}_3 \oplus \mathbf{p}_6$, $\boldsymbol{X}_3 = \mathbf{p}_4$, *and* $\boldsymbol{X}_4 = \mathbf{p}_5$ *are transmitted in this order. The receivers' decoding time* $\{u_{n,k}\}$ *are summarized in Table 2. The minimum APDD of this solution is* $D_{\mathcal{S}_s} = (1 \times 2 + 2 \times 5 + 3 \times 2 + 4 \times 3)/12 = 2.5$.

In each time slot of the coded transmission phase, the sender selects and broadcasts a coding set through erasure-prone wireless channels. We denote by U_T the block completion time of the coded transmission phase, and by D_T the APDD of this phase, calculated as in (1). U_T and D_T measure the throughput and decoding delay performance of this phase, respectively. They vary according to the IDNC solutions, transmission schemes, and erasure patterns. But it always holds that $U_T \geqslant U_s$ and $D_T \geqslant D_s$ if S-IDNC is applied. Therefore, U_s and D_s reflect the performance limits of S-IDNC. Hence, we will first study these limits in the next section, and then design S-IDNC transmission schemes and coding algorithms in Sections 4 and 5, respectively.

3 Performance limits and properties of IDNC

In this section, we study performance limits and properties of S-IDNC and compare it with G-IDNC.

Table 2 The decoding delay of original data packets at the receivers

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$
R_1	1	0	0	0	4	2
R_2	0	1	0	0	0	2
R_3	0	0	2	3	4	0
R_4	0	0	0	3	0	2
R_5	0	0	2	0	4	0

3.1 Absolute minimum block completion time U_s

We first study the throughput limit of S-IDNC, measured by the absolute minimum block completion time U_s. It has been proved that U_s is equal to the size of the minimum clique partition solution[3] of $\mathcal{G}_s$ [13], denoted by $\mathcal{S}_c$. This equivalence holds because of the following property:

Property 1. *Removing any vertex from the S-IDNC graph does not change the connectivity of the remaining vertices.*

This property holds because vertices in $\mathcal{G}_s$ represent different data packets. Thus, to remove all vertices from $\mathcal{G}_s$ (i.e., to complete the broadcast), at least $|\mathcal{S}_c|$ cliques must be removed, which yields $U_s = |\mathcal{S}_c|$.

According to graph theory, $|\mathcal{S}_c|$ is equal to the chromatic number[4] $\chi(\overline{\mathcal{G}}_s)$ of the complementary graph $\overline{\mathcal{G}}_s$, which has the same vertex set as $\mathcal{G}_s$, but has opposite vertex connectivity. We thus have $U_s = \chi(\overline{\mathcal{G}}_s)$. Bounds and approximations on the chromatic number of a given graph have been well-studied in the graph theory literature [33–35]. They provide some insights into the U_s of a given SFM. In this subsection, we are interested in the probabilistic characterization of U_s, as $\mathcal{G}_s$ is the consequence of random packet erasures in the systematic transmission phase. Specifically, we address the following question: *what is the relation between* U_s *and system parameters, including the number of data packets and receivers, as well as the packet erasure probability?*

For wireless broadcast, a common assumption on random packet erasures is that they are independently and Bernoulli distributed at each receiver R_n with an erasure probability of $P_{e,n}$. Under this assumption, a similar question has already been introduced and answered for the RLNC technique. RLNC has been proved to be asymptotically throughput-optimal, for each RLNC coded packet is almost surely linearly independent of the previous RLNC coded packet(s) when the finite field is sufficiently large [11]. Hence, RLNC is able to offer the smallest block completion time among all NC techniques. It has also been shown in [6, 36, 37] that the block completion time of RLNC scales as $\mathcal{O}(\ln(N))$ when K is a constant. Consequently, the throughput of RLNC vanishes with increasing number of receivers N. To prevent zero throughput, it has been proved in [38] that K should scale faster than $\ln(N)$.

Since the throughput of RLNC is already optimal, it cannot be exceeded by the throughput of S-IDNC. Hence, we can infer that the throughput of S-IDNC should also follow a vanishing behavior with increasing N. However, its rate and specific dependence on system parameters have not been fully characterized in the literature. In this subsection, we answer this question through the following theorem:

Theorem 1. *The mean of the absolute minimum block completion time U_s is a function of the block size K, the number of receivers N, and packet erasure probability $\{P_{e,n}\}_{n=1}^{N}$:*

$$E[\,U_s] = -K\left(\frac{1}{2}+o(1)\right)\sum_{n=1}^{N}\log_K\left(1-P_{e,n}^2\right), \qquad (2)$$

where $o(1)$ is a small term that approaches zero with increasing K.

Proof. Our approach is to model the complementary S-IDNC graph $\overline{\mathcal{G}}_s$ after the systematic phase as a random graph with i.i.d. edge generating probability. Recall that two vertices in $\overline{\mathcal{G}}_s$ are connected if the two data packets conflict, i.e., if at least one receiver has missed both packets. Therefore, the generating probability of every edge, denoted by P_c, is calculated as:

$$P_c = 1-\prod_{n=1}^{N}\left(1-P_{e,n}^2\right). \qquad (3)$$

Then, the key is to prove that different edges are generated independently. We first consider the independence between two adjacent edges. Without loss of generality let us consider the generation of $\boldsymbol{e}_{1,2}$ and $\boldsymbol{e}_{1,3}$, two edges that are adjacent via $\boldsymbol{v}_1$, and are incident to $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, respectively. We denote by $P(\boldsymbol{e}_{1,2})$ the probability that $\boldsymbol{e}_{1,2}$ is generated. It holds that $P(\boldsymbol{e}_{1,2}) = P(\boldsymbol{e}_{1,3}) = P_c$ in (3). We further denote by $P(\boldsymbol{e}_{1,2}|\boldsymbol{v}_1)$ the probability that $\boldsymbol{e}_{1,2}$ is generated conditioned on that $\boldsymbol{v}_1$ is generated. We then argue the following relations:

1. $P(\boldsymbol{e}_{1,2},\boldsymbol{v}_1) = P(\boldsymbol{e}_{1,2})$, because the generating of $\boldsymbol{e}_{1,2}$ already indicates that $\boldsymbol{v}_1$ is generated. Similarly, we also have $P(\boldsymbol{e}_{1,3},\boldsymbol{v}_1) = P(\boldsymbol{e}_{1,3})$;
2. $P(\boldsymbol{v}_1|\boldsymbol{e}_{1,2},\boldsymbol{e}_{1,3}) = 1$ because of the same reason as above;
3. $P(\boldsymbol{e}_1,\boldsymbol{e}_2|\boldsymbol{v}_1) = P(\boldsymbol{e}_1|\boldsymbol{v}_1)\cdot P(\boldsymbol{e}_2|\boldsymbol{v}_1)$, because if $\boldsymbol{v}_1$ is already generated, the generating of $\boldsymbol{e}_1$ (resp. $\boldsymbol{e}_2$) only depends on whether $\mathbf{p}_2$ (resp. $\mathbf{p}_3$) is wanted by some of the receivers who want $\mathbf{p}_1$. Since wanting $\mathbf{p}_2$ and $\mathbf{p}_3$ are independent events for every receiver, the generating of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ is independent conditioned on that $\boldsymbol{v}_1$ is generated;
4. $P(\boldsymbol{v}_1)^2 \approx P(\boldsymbol{v}_1) \approx 1$, and the accuracy increases quickly with increasing number of receivers N. This is because $P(\boldsymbol{v}_1)$ is the probability that at least one receiver has missed $\mathbf{p}_1$ in the systematic transmission phase. It has a value of $1-\prod_{n=1}^{N}(1-P_{e,n})$, which quickly approaches to 1 with increasing N.

Then, to prove that $\boldsymbol{e}_{1,2}$ and $\boldsymbol{e}_{1,3}$ are generated independently, we only need to show that $P(\boldsymbol{e}_{1,2},\boldsymbol{e}_{1,3}) = P(\boldsymbol{e}_{1,2})\cdot P(\boldsymbol{e}_{1,3})$:

$$\begin{aligned} P(\boldsymbol{e}_{1,2},\boldsymbol{e}_{1,3}) &= \frac{P(\boldsymbol{e}_{1,2},\boldsymbol{e}_{1,3}|\boldsymbol{v}_1)\cdot P(\boldsymbol{v}_1)}{P(\boldsymbol{v}_1|\boldsymbol{e}_{1,2},\boldsymbol{e}_{1,3})} \qquad (4)\\ &= P(\boldsymbol{e}_{1,2}|\boldsymbol{v}_1)\cdot P(\boldsymbol{e}_{1,3}|\boldsymbol{v}_1)\cdot P(\boldsymbol{v}_1)\\ &\approx P(\boldsymbol{e}_{1,2}|\boldsymbol{v}_1)\cdot P(\boldsymbol{e}_{1,3}|\boldsymbol{v}_1)\cdot P(\boldsymbol{v}_1)^2\\ &= P(\boldsymbol{e}_{1,2},\boldsymbol{v}_1)\cdot P(\boldsymbol{e}_{1,3},\boldsymbol{v}_1)\\ &= P(\boldsymbol{e}_{1,2})\cdot P(\boldsymbol{e}_{1,3}), \end{aligned}$$

where (4) follows Bayes' rule. Hence, the generation of $\boldsymbol{e}_{1,2}$ and $\boldsymbol{e}_{1,3}$ are asymptotically independent of each other.

On the the other hand, it is intuitive that two disjoint edges in $\overline{\mathcal{G}}_s$ are generated independently. Therefore, we can assume that all edges in $\overline{\mathcal{G}}_s$ are generated independently.

Consequently, $\overline{\mathcal{G}}_s$ can be modeled as an Erdõs-Rényi random graph [39], which has K vertices and i.i.d. edge generating probability of P_c. Figure 3 compares the mean number of edges (with a value of $K(K-1)/2\cdot P_c$) of our proposed random graph model and the simulated average number of edges in $\overline{\mathcal{G}}_s$. Our model shows virtually no deviation under all considered values of N and K.

From graph theory, given K and P_c, almost every random graph $\overline{\mathcal{G}}_s$ has a chromatic number of [40]:

$$\chi(\overline{\mathcal{G}}_s) = \frac{K}{\log K}\left(\frac{1}{2}+o(1)\right)\log\frac{1}{1-P_c}. \qquad (5)$$

Since $U_s = \chi(\overline{\mathcal{G}}_s)$, the above value is the mean of U_s. By substituting (3) into (5) we obtain (2). □

Theorem 1 has the following important corollary:

Corollary 1. *The mean $E[U_s]$ of the absolute minimum block completion time of S-IDNC increases almost linearly with the number of receivers when all receivers experience similar packet erasure probabilities.*

Proof. This corollary can be proved by letting $\{P_{e,n}\}_{n=1}^{N} = P_e$, which will transform (2) into a linear function of N:

$$\begin{aligned} E[\,U_s] &= -K\left(\frac{1}{2}+o(1)\right)\sum_{n=1}^{N}\log_K\left(1-P_e^2\right)\\ &= -K\left(\frac{1}{2}+o(1)\right)\log_K\left(1-P_e^2\right)\cdot N. \qquad (6) \end{aligned}$$

□

Then, by noting that the mean block completion time of the coded transmission phase is lower bounded by $E[U_s]$, we conclude that the throughput of S-IDNC degrades with increasing number of receivers. Such degradation is common among network coding techniques that aim to

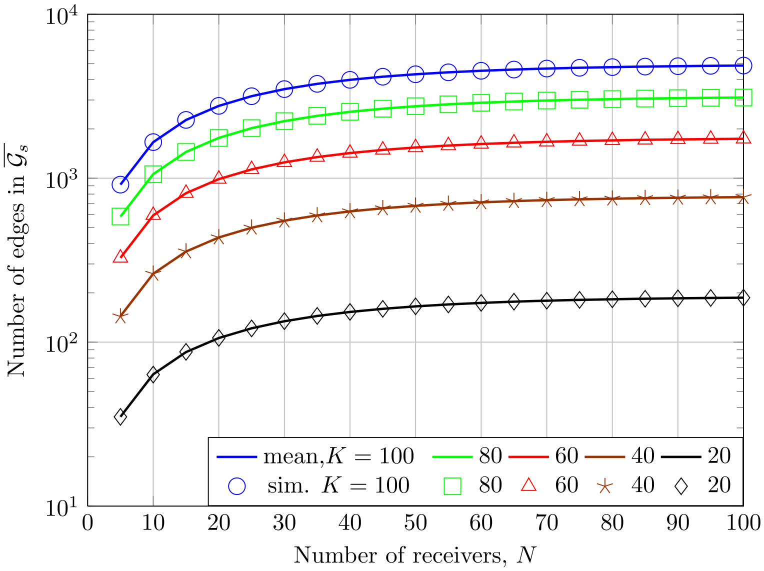


Fig. 3 The mean and simulated number of edges in $\overline{\mathcal{G}}_s$ when $\{P_{e,n}\}_{n=1}^{N} = 0.2$ and $K \in [20, 100]$

reduce decoding delay [41]. In the next subsection, we will study the APDD of S-IDNC.

3.2 Absolute minimum average packet decoding delay D_s

Unlike U_s, to the best of our knowledge, there is no existing hardness result on finding D_s. In this subsection, we address it through the following theorem and then propose an upper bound on D_s.

Theorem 2. *It is NP-hard to find the absolute minimum APDD D_s of S-IDNC.*

In order to prove it, we first reveal the perfect decoding scenario of each receiver. We first note that it is impossible for a receiver to decode more than u wanted data packets from the first u coded packets, for any $u > 0$. Then for every receiver R_n, its perfect decoding scenario is to decode one of its $|\mathcal{W}_n|$ wanted data packet from each of the first $|\mathcal{W}_n|$ coded packets. Intuitively, this scenario minimizes both the BCC and APDD experienced by R_n. By extending the perfect decoding scenario to all receivers, we obtain the concept of *perfect S-IDNC solution*:

Definition 3. *An S-IDNC solution is perfect and is denoted by $\mathcal{S}_p$ if every receiver R_n can decode one of its $|\mathcal{W}_n|$ wanted data packets from each of the first $|\mathcal{W}_n|$ coding sets in $\mathcal{S}_p$.*

By its definition, $\mathcal{S}_p$ offers the perfect packet decoding scenario for all receivers; every coded packet allows *all* receivers that are still missing data packets to decode one wanted data packet. Therefore, its APDD, denoted by $D_{\mathcal{S}_p}$, is a lower bound of D_s, and can only be achieved if $\mathcal{S}_p$ exists. The value of $D_{\mathcal{S}_p}$ is:

$$D_{\mathcal{S}_p} = \frac{1}{T} \sum_{n=1}^{N} \sum_{i=1}^{w_n} i. \tag{7}$$

The hardness of deciding the existence of $\mathcal{S}_p$ is as follows:

Theorem 3. *It is NP-complete to decide the existence of $\mathcal{S}_p$ for a given SFM.*

It is proved through a reduction from a graph γ-colorability ($\gamma \geqslant 3$) problem, which is well-known to be NP-complete [34]. The proof is given in Appendix 1. Then by noting that $D_{\mathcal{S}_p}$ can only be achieved by $\mathcal{S}_p$, Theorem 3 has the following corollary:

Corollary 2. *It is NP-complete to decide the achievability of $D_{\mathcal{S}_p}$ for a given SFM.*

Corollary 2 proves Theorem 2 by contradiction; if it is easy to find D_s for a given SFM, then we can easily decide the achievability of $D_{\mathcal{S}_p}$ by comparing D_s with $D_{\mathcal{S}_p}$, as $D_s = D_{\mathcal{S}_p}$ means that $D_{\mathcal{S}_p}$ is achievable, and $D_s > D_{\mathcal{S}_p}$ means that $D_{\mathcal{S}_p}$ is not achievable. However, this result contradicts with Corollary 2. Hence, it is NP-hard to find D_s.

Besides the NP-hardness, D_s has the following property:

Property 2. *The absolute minimum APDD D_s is upper bounded by U_s as*

$$D_s \leqslant \frac{U_s + 1}{2}. \tag{8}$$

Proof. Given an S-IDNC solution $\mathcal{S}_s = \{\mathcal{M}_u\}_{u=1}^{U}$, let $T(u) = \sum_{\mathbf{p}_k \in \mathcal{M}_u} T_k$ be the number of receivers who can decode a data packet from $\mathcal{M}_u$. The minimum APDD of $\mathcal{S}_s$ is thus:

$$D_{\mathcal{S}_s} = \frac{1}{T} \sum_{u=1}^{U} T(u) \cdot u. \tag{9}$$

which is maximized when $\{T(u)\}_{u=1}^{U} = \frac{T}{U}$. In this case, $D_{\mathcal{S}_s} = \frac{U+1}{2}$. Applying this result to an S-IDNC solution with absolute minimum block completion time $U = U_s$, we obtain the result. □

Our proof indicates that, although it is NP-hard to achieve D_s, we can still effectively reduce APDD by reducing the sizes of our S-IDNC solutions. Before we further explore this result to implement S-IDNC, we would like to compare the performance limits of S-IDNC that we have just derived with G-IDNC.

3.3 S-IDNC vs. G-IDNC

In this subsection, we address the question of *how does S-IDNC compare with G-IDNC?*

We first note that the NP-hardness of finding D_s also holds for D_g. This is because the perfect S-IDNC solution $\mathcal{S}_p$ is also the best possible G-IDNC solution. For the throughput, we first present a relation between S- and G-IDNC graphs (proved in the appendix):

Theorem 4. *The minimum clique partition solutions of S-IDNC and G-IDNC graphs have the same size. In other words, $\chi(\overline{\mathcal{G}}_s) = \chi(\overline{\mathcal{G}}_g)$.*

This theorem, together with Corollary 1, indicates that $\chi(\overline{\mathcal{G}}_g)$ also increases almost linearly with N when all receivers experience similar erasure probabilities. However, the above theorem does not imply $U_s = U_g$. This is because G-IDNC does not have Property 1. Explicitly, by removing a vertex from $\mathcal{G}_g$, more edges and larger cliques may be generated, and thus the absolute minimum block completion time U_g can be smaller than $\chi(\overline{\mathcal{G}}_g)$ of the original G-IDNC graph $\mathcal{G}_g$ [24]. We thus have $U_g \leqslant U_s$. We note, however, that a systematic way of finding U_g other than brute-force search remains widely open.

4 S-IDNC transmission schemes

In this section, we design S-IDNC transmission schemes to compensate for packet erasures in the coded transmission phase. To this end, the sender has to regularly collect feedback from the receivers about their packet reception state to make online coding decisions. We consider transmission schemes with two different types of feedback frequency, namely:

1. Fully-online feedback: feedback is collected after every coded transmission. However, this could be costly in wireless communications. We thus also consider a reduced feedback frequency next;
2. Semi-online feedback: feedback is only collected after transmitting a complete S-IDNC solution;

To be able to design S-IDNC transmission schemes, two questions need to be answered first:

1. What is the optimization objective for throughput and decoding delay improvement?
2. What does the sender need to send to achieve it?

Before addressing these questions, we first highlight some challenges:

Remark 1. *Under random packet erasures, a reasonable measure of throughput is the mean block completion time $E[U_T]$ of the coded transmission phase. However, it is intractable to minimize $E[U_T]$. To see this, let us consider the stochastic shortest path (SSP) method [23]. In SSP method, the state space comprises the current SFM and its successors, and thus has a prohibitively large size with a value of 2^T, where T is the number of "1"s in $\mathbf{A}$. The action space for each state comprises all cliques/coding sets, which is NP-hard to find [42]. Then, $E[U_T]$ is recursively minimized by examining all the states and the associated actions. Such examination is necessary, because the packet erasures can take any pattern and are not predictable. But it makes $E[U_T]$ intractable to minimize. To overcome this difficulty, we will turn to optimization objectives that are heuristic, but still based on SSP optimization principles.*

Remark 2. *It is intractable to minimize the APDD D_T of the coded transmission phase due to the NP-hardness of finding D_s, because otherwise by setting $P_e = 0$, the minimum D_T is equal to D_s. To overcome this difficulty, we will give higher priority to the minimization of block completion time. In other words, we first minimize the block completion time. Then among the resultant coding decisions, we choose the one that minimizes the decoding delay. Our prioritization reflects the motivation of using network coding, that is, to achieve better throughput performance. It also provides bounded decoding delay performance as we have shown in (8). This will also be confirmed by our* *simulations, which show that D_T generally decreases with decreasing U_T.*

4.1 Fully-online transmission scheme

In this scheme, the current state in SSP is the current SFM $\boldsymbol{A}$, the absorbing state is the all-zero SFM and is denoted by $\boldsymbol{A}_0$. The action space comprises all the S-IDNC coding sets of $\boldsymbol{A}$. The cost of each action is one, for it consumes one transmission. The block completion time U_T is thus equal to the number of transitions (a.k.a. path length or distance) between $\boldsymbol{A}$ and $\boldsymbol{A}_0$.

According to Remark 1, it is intractable to choose an action/coded packet that minimizes the expected path length (and thus $E[U_T]$). As a heuristic alternative, we propose to choose an action/coded packet that belongs to the shortest path from $\boldsymbol{A}$ to $\boldsymbol{A}_0$, which has a length of U_s. This choice guarantees that, upon the reception of the coded packet at all interested receivers, the shortest distance between the updated state $\boldsymbol{A}'$ and $\boldsymbol{A}_0$ is minimized to $U_s - 1$. To this end, the coded packet must belong to a minimum clique partition solution $\mathcal{S}_c$. Otherwise, the shortest distance between $\boldsymbol{A}'$ and $\boldsymbol{A}_0$ is still U_s.

We then reduce APDD by forcing the coded packet to be maximal (and thus serving the maximal number of receivers). However, cliques in a minimum clique partition solution are not necessarily maximal. Hence, we further require the coded packet to belong to a set of U_s maximal cliques that together cover all the data packets. This set is also an S-IDNC solution and is denoted by $\mathcal{S}_m$.

In conclusion, we propose the following coded packet $\mathcal{M}_f$ for fully-online transmission scheme:

Given an SFM instance, the preferred coded packet $\mathcal{M}_f$ is the most wanted coded packet in $\mathcal{S}_m$, where $\mathcal{S}_m$ is an S-IDNC solution that contains U_s maximal cliques.

4.2 Semi-online transmission scheme

The current and absorbing states in this scheme is the same as in the fully-online scheme. But the action space becomes the set of all S-IDNC solutions $\mathbb{S}_s$, and the cost of each action is the solution size $|\mathcal{S}_s|$, which is equal to the length of a semi-online transmission round. The total cost is thus equal to the block completion time.

According to Remark 1, it is intractable to minimize the expected total cost (and thus $E[U_T]$). As a heuristic alternative, we propose to minimize the expected cost of the shortest path between $\boldsymbol{A}$ and $\boldsymbol{A}_0$. The shortest path includes only one transition, representing the event that every coded packet of the chosen solution $\mathcal{S}_s$ is received by all the interested receivers after only one semi-online round. Denote the probability of this event by P_s. Then the expected cost is $|\mathcal{S}_s|/P_s$, where P_s is calculated as:

$$P_s = \prod_{k=1}^{K} \prod_{R_n \in \mathcal{T}_k} \left(1 - P_{e,n}^{d_k}\right). \tag{10}$$

Here, d_k is called the packet multiplicity and is defined below.

Definition 4. *The multiplicity d_k of data packet $\mathbf{p}_k$ is the number of coding sets in $\mathcal{S}_s$ that comprise $\mathbf{p}_k$.*

We note that the minimum clique partition solution $\mathcal{S}_c$ is not a preferred semi-online S-IDNC solution. Although $\mathcal{S}_c$ offers the smallest solution size ($|\mathcal{S}_c| = U_s$), it does not maximize P_s because every data packet has a multiplicity of only one due to disjoint cliques in $\mathcal{S}_c$. In contrast, the $\mathcal{S}_m$ we have proposed for the fully-online case can offer a higher P_s than $\mathcal{S}_c$ due to possibly overlapping maximal cliques, while also offering the smallest solution size.

We still wish to answer the following question before choosing $\mathcal{S}_m$ as our preferred semi-online S-IDNC solution: *Is there a solution that, though is large in its size, provides higher packet multiplicities, so that P_s is maximized?*

An explicit answer to this question is difficult to obtain, because it requires the examination of all the solutions of size greater than U_s. Such search is costly and does not provide any insight into this question. Moreover, a larger solution is unlikely to provide higher packet multiplicities due to the following property of S-IDNC solutions:

Property 3. *Every coding set in an S-IDNC solution comprises at least one data packet with a multiplicity of one.*

This property holds because if every data packet in a coding set has a multiplicity of greater than one, then this coding set can be removed from the solution without affecting the completeness of the solution. Due to the above property, an S-IDNC solution $\mathcal{S}_s$ has at least $|\mathcal{S}_s|$ data packets with a multiplicity of only one. According to (10), these unit-multiplicity data packets reduce P_s the most. Hence, S-IDNC solutions with a larger size may have more unit-multiplicity data packets than $\mathcal{S}_m$, and thus are not preferable.

Therefore, we choose $\mathcal{S}_m$ for throughput improvement. Then, by taking into account our secondary optimization objective, i.e., the APDD, we define our preferred semi-online S-IDNC solution as follows:

Given an SFM instance, the preferred semi-online S-IDNC solution is $\mathcal{S}_m$, which comprises a set of U_s maximal cliques. The cliques are sorted for transmission in the descending order of their numbers of targeted receivers to minimize the APDD.

A flow-chart of the proposed two transmission schemes are presented in Fig. 4. Both the fully- and semi-online IDNC schemes require finding $\mathcal{S}_m$. Since packet multiplicity is not a concern in graph theory, algorithms that find $\mathcal{S}_m$ do not exist in the graph theory literature. Hence, we will design algorithms dedicated for S-IDNC in the next section. Before moving on, we briefly compare S-IDNC and G-IDNC under the above two transmission schemes.

4.3 S-IDNC vs. G-IDNC

With fully-online feedback, the sender can update the G-IDNC graph $\mathcal{G}_g$ and add new edges representing coding opportunities after every transmission. The throughput of G-IDNC is thus better than S-IDNC. But, the price is high computational load, because G-IDNC graph is much larger than S-IDNC graph ($\mathcal{O}(NK)$ vs. $\mathcal{O}(K)$). On the other hand, it has been proved in [30] that, when receiver feedback is not available, the best strategy for the sender is not to update $\mathcal{G}_g$ (compared with certain probabilistic update strategy). Therefore, during a semi-online transmission round, the sender only sends the minimum clique partition solution of $\mathcal{G}_g$, which, according to Theorem 4, has the same size as the minimum clique partition solution of S-IDNC.

5 S-IDNC coding algorithms

The two transmission schemes we proposed in the last section require finding $\mathcal{S}_m$, an S-IDNC solution that contains U_s maximal coding sets. In this section, we develop its optimal and heuristic algorithms.

5.1 Optimal S-IDNC coding algorithm

Our optimal algorithm finds all valid $\mathcal{S}_m$ in two steps:

Step-1 *Find all the maximal coding sets (maximal cliques):* This problem is NP-hard in graph theory [42], and thus cannot be optimally solved by an algorithm whose computational complexity is a polynomial of the number K of data packets. However, it can be solved by exponential algorithms such as Bron-Kerbosch (B-K) algorithm [42]. We use B-K algorithm to optimally find the group of all maximal cliques and denote the group by $\mathcal{A}$.

Step-2 *Find all valid $\mathcal{S}_m$ from $\mathcal{A}$:* We propose a branching algorithm in Algorithm 1. The intuition behind this algorithm is that, if a data packet $\mathbf{p}_k$ belongs to d_k maximal coding sets in $\mathcal{A}$, then one of these d_k maximal coding sets must be included in $\mathcal{S}_m$ for the completeness of $\mathcal{S}_m$. In the extreme case where $d_k = 1$, the sole maximal coding set that contains $\mathbf{p}_k$ must be included in $\mathcal{S}_m$. Below is an example of Algorithm 1.

Example 4. *Consider the graph model in Fig. 5. In Step-1, we find all the maximal cliques:* $\mathcal{A} = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5), (\mathbf{p}_3, \mathbf{p}_4), (\mathbf{p}_4, \mathbf{p}_6), (\mathbf{p}_5, \mathbf{p}_6)\}$. *Then in Step-2:*

1. *Initially,* $\mathcal{S} = \emptyset$, $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S} = \mathcal{A}$, *and the set of data packets not included in* $\mathcal{S}$ *is* $\overline{\mathcal{P}} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6\}$. *Since* $\mathbf{p}_1$ *is only included in* $(\mathbf{p}_1, \mathbf{p}_3)$ *and* $\mathbf{p}_2$ *is only included in* $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)$, *these two coding sets must be added to* $\mathcal{S}$. *Hence,* $\mathcal{S} = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)\}$ *after the first two iterations;*
2. *The set of data packets not included in* $\mathcal{S}$ *is* $\overline{\mathcal{P}} = \{\mathbf{p}_4, \mathbf{p}_6\}$, *and the remaining maximal coding sets are* $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S}$=$\{(\mathbf{p}_3, \mathbf{p}_4), (\mathbf{p}_4, \mathbf{p}_6), (\mathbf{p}_5, \mathbf{p}_6)\}\}$. *Since* $\mathbf{p}_4$ *has a multiplicity of 2 under* $\overline{\mathcal{S}}$ *due to* $(\mathbf{p}_3, \mathbf{p}_4)$ *and* $(\mathbf{p}_4, \mathbf{p}_6)$, *we branch* $\mathcal{S}$ *into two successors:* $\mathcal{S}_1 = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5), (\mathbf{p}_4, \mathbf{p}_5)\}$ *and* $\mathcal{S}_2 = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5), (\mathbf{p}_4, \mathbf{p}_6)\}$. *Since* $\mathcal{S}_2$ *contains all data packets and there are no other branching opportunities, the algorithm stops and outputs* $\mathcal{S}_2$ *as* $\mathcal{S}_m$.

B-K algorithm and Algorithm 1 constitute our optimal S-IDNC coding algorithm. It is optimal because it exhaustively finds all the valid $\mathcal{S}_m$ from all the maximal

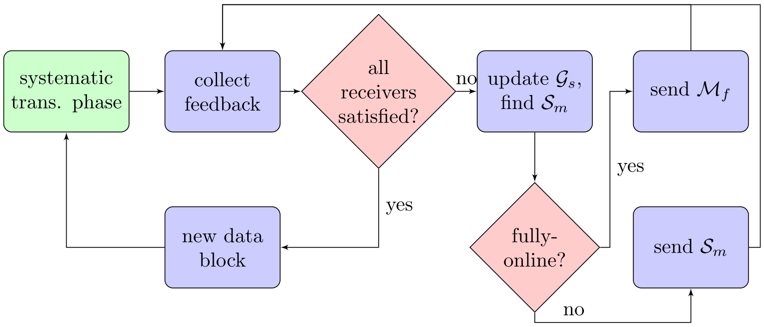


Fig. 4 The fully- and semi-online transmission schemes

Algorithm 1 Optimal S-IDNC solution search

```
input: the group of all maximal coding sets, $\mathcal{A}$;
initialization: a set $\mathcal{B}$ of solutions, $\mathcal{B}$ only contains an empty solution $\mathcal{S} = \emptyset$. A counter $u = 1$;
while no solution in $\mathcal{B}$ contains all data packets, do
    while there is a solution in $\mathcal{B}$ with size $u - 1$, do
        Denote this solution by $\mathcal{S} = \{\mathcal{M}_1, \cdots, \mathcal{M}_{u-1}\}$. Denote the data packets included in $\mathcal{S}$ by $\mathcal{P} = \bigcup_{i=1}^{u-1}\{\mathcal{M}_i\}$ and all data packets not included in $\mathcal{S}$ by $\overline{\mathcal{P}} = \mathcal{P}_K \setminus \mathcal{P}$. Also denote the maximal coding sets not included in $\mathcal{S}$ by $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S}$;
        Pick from $\overline{\mathcal{P}}$ the data packet $\mathbf{p}$ that has the smallest multiplicity $d$ under $\overline{\mathcal{S}}$. Denote the $d$ coding sets which contain $\mathbf{p}$ by $\mathcal{M}'_1, \cdots, \mathcal{M}'_d$;
        Branch $\mathcal{S}$ into $d$ new solutions, $\mathcal{S}'_1, \cdots, \mathcal{S}'_d$. Then, add $\mathcal{M}'_1, \cdots, \mathcal{M}'_d$ to these solutions, respectively. The sizes of the new solutions are $u$;
    end while
    $u = u + 1$;
end while
Output the solutions in $\mathcal{B}$ that contain all data packets.
```

coding sets. Among these solutions, we can choose the one that optimizes a secondary criteria, such as the one offering the smallest $D_{\mathcal{S}}$, or the largest $P_{\mathcal{S}}$, calculated using (10).

5.2 Hybrid S-IDNC coding algorithm

Algorithm 1 is memory demanding, because the number of candidate solutions grows exponentially with branching. Thus, we propose a heuristic alternative to it. The idea is to iteratively maximize the number of data packets included in $\mathcal{S}_m$. The algorithm is given in Algorithm 2.

B-K algorithm and Algorithm 2 constitute our hybrid S-IDNC coding algorithm. It produces only one S-IDNC solution, with no guarantee on the solution size. It is still computational expensive due to B-K algorithm. Thus, we develop a polynomial time heuristic S-IDNC coding algorithm next.

Algorithm 2 Hybrid S-IDNC solution search

```
input: the group of all maximal coding sets, $\mathcal{A}$;
initialization: an empty solution $\mathcal{S} = \emptyset$, a counter $u = 1$, packet set $\mathcal{P} = \mathcal{P}_K$;
while $\mathcal{S}$ does not contain all data packets, do
    find the coding set $\mathcal{M}$ in $\mathcal{A}$ that contains the largest number of data packets in $\mathcal{P}$;
    Add $\mathcal{M}$ to $\mathcal{S}$ and remove data packets in $\mathcal{M}$ from $\mathcal{P}$;
    $u = u + 1$;
end while
Output the solution $\mathcal{S}$.
```

5.3 Heuristic S-IDNC coding algorithm

Algorithm 3 is a simple algorithm that heuristically finds the maximum (the largest maximal) clique of a graph. The intuition behind this algorithm is that, a vertex is very likely to be in the maximum clique if it is incident by the largest number of edges. Variations of this algorithm have been developed in the literature [12, 13, 23]. But, this algorithm has not been applied to finding a complete S-IDNC solution, and its computational complexity has not been identified yet.

Algorithm 3 Heuristic maximum clique search

```
input: graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$;
initialization: an empty vertex set $\mathcal{V}_{\text{keep}}$;
while $\mathcal{G}$ is not empty do
    add to $\mathcal{V}_{\text{keep}}$ the vertex $\boldsymbol{v}$ which has the largest number of edges incident to it;
    update $\mathcal{G}$ by deleting $\boldsymbol{v}$ and all the vertices not connected to $\boldsymbol{v}$ (These vertices can be ignored because they cannot be part of the target clique, which contains $\boldsymbol{v}$);
end while
vertices in $\mathcal{V}_{\text{keep}}$ are pair-wise connected, and no other vertices can be added to them. Hence, $\mathcal{V}_{\text{keep}}$ is a maximal clique.
```

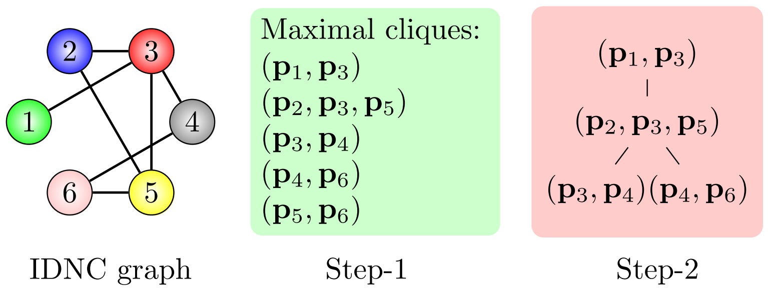

2
3
1
4
6
5
Maximal cliques:
$(\mathbf{p}_1, \mathbf{p}_3)$
$(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)$
$(\mathbf{p}_3, \mathbf{p}_4)$
$(\mathbf{p}_4, \mathbf{p}_6)$
$(\mathbf{p}_5, \mathbf{p}_6)$
$(\mathbf{p}_1, \mathbf{p}_3)$
$(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)$
$(\mathbf{p}_3, \mathbf{p}_4)(\mathbf{p}_4, \mathbf{p}_6)$
IDNC graph
Step-1
Step-2

Fig. 5 An example of the two-step optimal S-IDNC coding algorithm

The computational complexity of Algorithm 3 is polynomial in the number of data packets K. The highest computational cost occurs when the input graph is complete, i.e., when all vertices are connected to each other. In this case, only one vertex will be removed in each iteration. Thus, the number of remaining vertices in iteration-i will be $K - i$, $\forall i \in [0, K - 1]$. Then, to find the vertex with the largest number of incident edges, we need $K - i$ comparisons. The total computational cost is thus in the order of $\sum_{i=0}^{K-1} K - i = K(K-1)/2$. Hence, the computational complexity of Algorithm 3 is at most $\mathcal{O}(K^2)$.

We apply Algorithm 3 to iteratively find $\mathcal{S}_m$ in Algorithm 4. In each iteration, we find a clique using Algorithm 3, maximize it by adding more vertices to it whenever possible, and then remove it from the S-IDNC graph. This will increase the multiplicities of the added vertices/packets. Below is an example:

Example 5. *Consider the graph $\mathcal{G}_s$ in Fig. 1b. In the first two iterations, the algorithm will choose $\mathcal{M}_1 = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$ and $\mathcal{M}_2 = (\mathbf{p}_3, \mathbf{p}_6)$, respectively. In the third iteration, $\mathcal{V}_{covered} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_6\}$ and the algorithm can only choose $\mathcal{M}_3 = (\mathbf{p}_5)$. Among all the data packets in $\mathcal{V}_{covered}$, $\mathbf{p}_2$ can be added to $\mathcal{M}_3$. Thus $\mathcal{M}_3 = \{\mathbf{p}_2, \mathbf{p}_5\}$. The algorithm then stops and outputs $\mathcal{S}_m = \{(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4), (\mathbf{p}_3, \mathbf{p}_6), (\mathbf{p}_2, \mathbf{p}_5)\}$.*

Algorithm 4 Heuristic S-IDNC solution search

```
input: a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$;
initialization: an empty vertex set $\mathcal{V}_{\text{covered}}$, a working
  graph $\mathcal{G}_w = \mathcal{G}$, and a counter $i = 0$;
while $\mathcal{V}_{\text{covered}} \neq \mathcal{V}$ do
    Find the maximum clique in $\mathcal{G}_w$ using Algorithm 3.
      Denote it by $\mathcal{M}_i$ ;
    Find the vertices in $\mathcal{V}_{\text{covered}}$ which are connected to
      $\mathcal{M}_i$. Denote their set by $\mathcal{V}_i$ (They are the candidate
      vertices that could be added to $\mathcal{M}_i$.);
    Generate a subgraph of $\mathcal{G}$ whose vertex set is $\mathcal{V}_i$.
      Denoted this subgraph by $\mathcal{G}'_i(\mathcal{V}_i, \mathcal{E}_i)$;
    Find the maximum clique in $\mathcal{G}'_i$ using Algorithm 1,
      denoted it by $\mathcal{M}'_i$ (All vertices in $\mathcal{M}'_i$ are connected
      to each other and thus can all be added to $\mathcal{M}_i$.);
    Update $\mathcal{V}_{\text{covered}}$ by adding vertices in $\mathcal{M}_i$ into it;
    Update $\mathcal{G}_w$ by removing $\mathcal{M}_i$ from it;
    Update $\mathcal{M}_i$ as $\mathcal{M}_i = \mathcal{M}_i \cup \mathcal{M}'_i$ (The new clique is
      at least as large as the old one, and thus provides
      higher packet multiplicity);
    $i = i + 1$;
end while
```

In conclusion, we proposed an optimal coding algorithm that exhaustively finds all the possible $\mathcal{S}_m$, and also proposed its hybrid and heuristic alternatives. Both the optimal and hybrid algorithms are exponential-time algorithms in terms of K due to their use of B-K algorithm, while the heuristic algorithm is a polynomial-time one. The output $\mathcal{S}_m$ is used as the S-IDNC solution for the semi-online transmission scheme. If fully-online transmission scheme is applied, the transmitted coding set $\mathcal{M}_f$ is chosen from $\mathcal{S}_m$.

6 Simulations

In this section, we present the simulated throughput and decoding delay performance of S-IDNC (abbreviated as S- in the figures) under different scenarios, including under fully- and semi-online transmission schemes, and under the use of optimal, hybrid and heuristic coding algorithms (abbreviated in the figures as Fully-, Semi-, Opt., Hybr., and Heur., respectively).

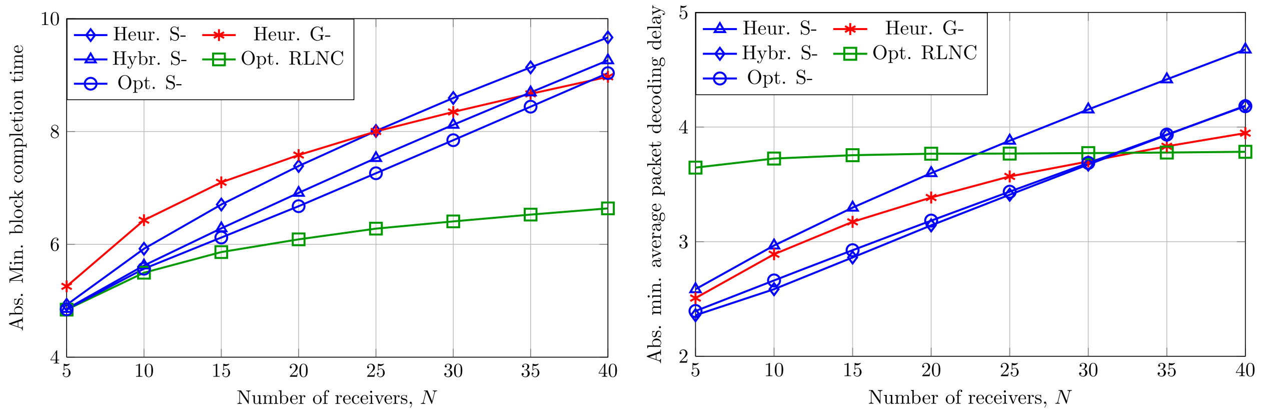
We also compare S-IDNC with RLNC and G-IDNC. For RLNC, we assume a sufficiently large finite field, so that its throughput is almost surely optimal and serves as a benchmark. For G-IDNC, although its best performance is at least as good as S-IDNC (as we have explained in Section 3.3), this advantage will not necessarily be reflected in our simulation results. This is because there has not been any optimal G-IDNC algorithm. Instead, we apply a heuristic algorithm (abbreviated as Heur. G- in the figures) proposed in [23], which aims at minimizing the block completion time. This aim coincides with our optimization priorities for S-IDNC in Remark 2, namely, to minimize the block completion time first.

We conduct four sets of simulations. In all simulations we apply a block size of $K = 15$. In the first three sets, we fix $P_e = 0.2$ and set the number of receivers $N \in [5 and 40]$. In the fourth set, we fix $N = 15$ and set $P_e \in [0.05, 0.4]$.

The purposes of the four sets of simulations are as follows. The first set compares the performance limits of the three techniques. The results are presented in Fig. 6. The second (resp. third) set of simulations compares the throughput and decoding delay performance under fully-online (resp. semi-online) transmission schemes. The results are presented in Fig. 7 (resp. Fig. 8). We note that the performance of RLNC is the same under both schemes, because RLNC is feedback-free. In the fourth set, we evaluate the performance of our hybrid algorithm under different packet erasure probabilities and compare it with fully-online heuristic G-IDNC and RLNC. The results are presented in Fig. 9.

Our observations on S-IDNC are as follows:

- The absolute minimum block completion time of S-IDNC increases almost linearly with N. This result matches Corollary 1;
- The fully-online transmission scheme always provides better throughput and decoding delay performance than the semi-online one;



(a) Block completion time

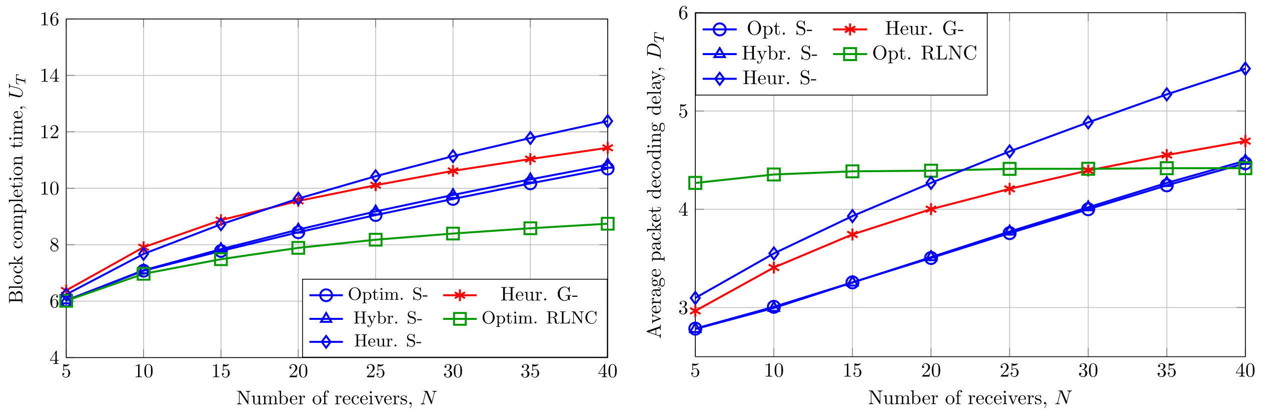
(b) Average packet decoding delay

Fig. 6 The throughput and decoding delay performance limits of S- and G-IDNC, and RLNC

- The optimal coding algorithm always provides better throughput performance than its hybrid and heuristic alternatives. This result verifies our choice of $\mathcal{S}_m$ for throughput improvement, because only the optimal coding algorithm can always produce $\mathcal{S}_m$, which has $|\mathcal{S}_m| = U_s$;
- However, the optimal coding algorithm does not necessarily minimize the APDD. For example, in Fig. 6b, the hybrid algorithm provides slightly smaller APDD than the optimal one when there are no packet erasures and when the number of receivers is $N \leqslant 15$;
- The performance gap between the optimal and hybrid algorithms is always marginal, and is much smaller than their gap with the heuristic one. Hence, the hybrid algorithm strikes a good balance between performance and computational load.

A cross comparison of RLNC, S-, and G-IDNC shows that:

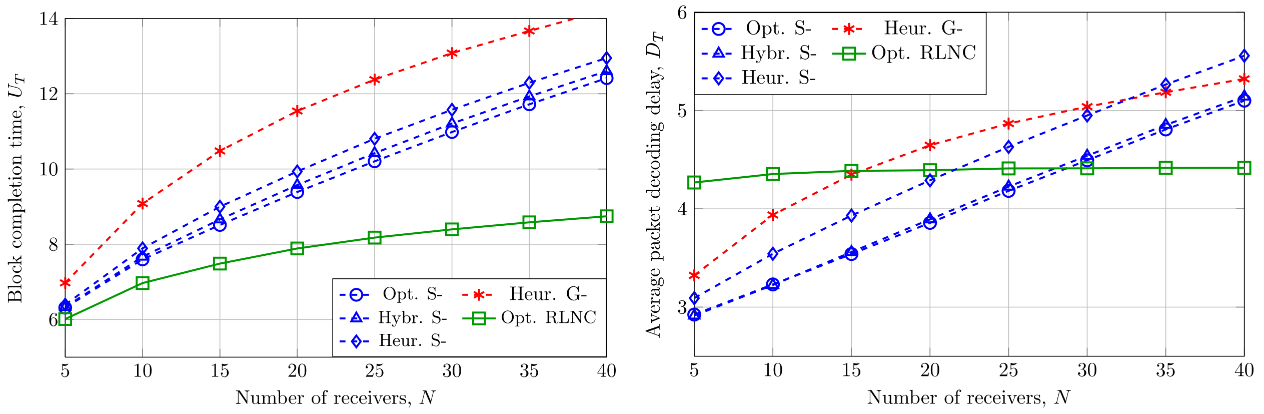
- The throughput of RLNC is always the best. The throughput of S-IDNC is very close to RLNC when the number of receivers is small. Their gap increases with N;
- In general, the APDD of both S- and G-IDNC is better than RLNC. This advantage only vanishes when the block completion time of S- and G-IDNC becomes much larger than RLNC, which takes place when N is much larger than K;



(a) Block completion time

(b) Average packet decoding delay

Fig. 7 The throughput and decoding delay performance of fully online transmission scheme when different coding algorithms are applied. It is compared with the performance of heuristic fully-online G-IDNC and RLNC



(a) Block completion time

(b) Average packet decoding delay

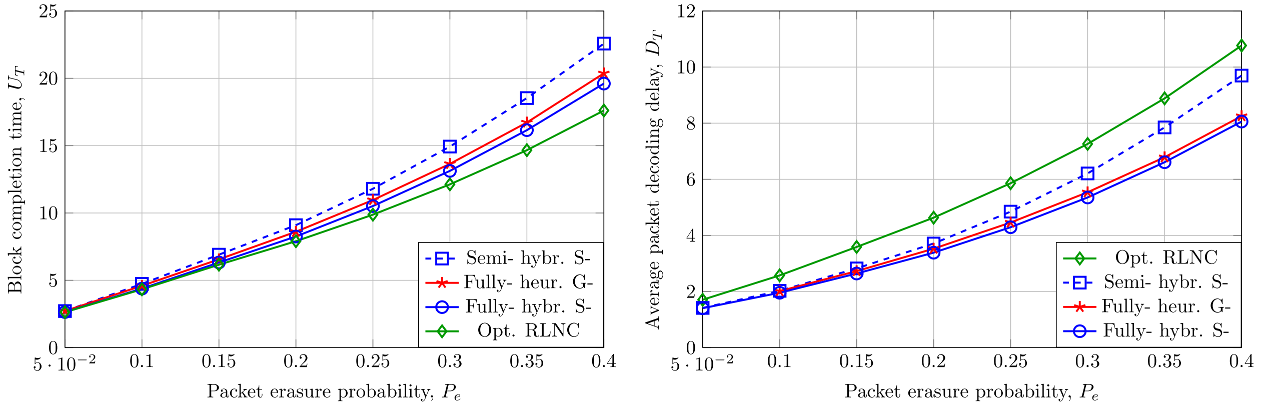
Fig. 8 The throughput and decoding delay performance of semi-online transmission scheme when different coding algorithms are applied. It is compared with the performance of heuristic semi-online G-IDNC and RLNC

- With a moderate number of receivers $N = 15$, the APDD of S-IDNC is better than RLNC under all simulated values of packet erasure probability;
- There is no clear winner between the performance of heuristic G-IDNC and optimal S-IDNC. We can expect that G-IDNC will outperform S-IDNC if its optimal coding algorithm is developed.

In summary, our simulations verified our theorems, propositions, and algorithms. They also demonstrated that, if we are concerned with both throughput and decoding delay performance, S-IDNC is a good alternative to RLNC when the number of receivers is not too large.

7 Conclusions

In this paper, we studied the throughput and average packet decoding delay (APDD) performance of S-IDNC in broadcasting a block of data packets to wireless receivers under packet erasures. By using a random graph model, we showed that the throughput of S-IDNC decreases with increasing an number of receivers. By introducing the concept of perfect S-IDNC solution, we proved the NP-hardness of APDD minimization. We derived an upper bound on APDD and showed that minimizing the IDNC solution size can effectively reduce APDD. By applying stochastic shortest path method, we showed that it is intractable to make optimal coding decisions in the presence of random packet erasures. We then



(a) Block completion time

(b) Average packet decoding delay

Fig. 9 The throughput and decoding delay performance of hybrid S-IDNC with different feedback frequencies and under different P_e. It is compared with the performance of heuristic fully-online G-IDNC and RLNC

used heuristic objective functions to determine the preferred coded packet(s) to send when fully- or semi-online receiver feedback is collected. We developed optimal and heuristic S-IDNC coding algorithms that minimize the solution size and increase packet multiplicity. We also compared S-IDNC with G-IDNC by proving the equivalence between the chromatic number of the complementary S-IDNC and G-IDNC graphs.

Our work provides new understandings of S-IDNC. It will facilitate the extension of S-IDNC to applications in other network settings, such as cooperative data exchange and distributed data storage. We are also interested in designing approximation and heuristic algorithms for APDD minimization.

Endnotes

[1] We assume that there exists an error-free feedback link from each receiver to the sender that can be used with an appropriate frequency. The feedback content is the index of the data packets it has received. Each index has a length of $\log_2 K$ bits. The feedback frequency will be studied in Section 4.

[2] In traditional G-IDNC, non-instantly decodable packets will be discarded by the receivers. Storing such packets may further improve throughput with extra computational cost. This problem is beyond the scope of this paper. Interested readers are referred to [32] for a recent treatment.

[3] The minimum clique partition solution of a graph $\mathcal{G}$ is the minimum set of disjoint cliques of $\mathcal{G}$ that together cover all the vertices.

[4] The chromatic number of a graph $\mathcal{G}$ is the minimum number of colors to color the vertices so that any two connected vertices have different colors.

Appendix 1: Proof of Theorem 3

In this appendix we prove that it is NP-complete to decide the existence of the perfect S-IDNC solution $\mathcal{S}_p$ for a given SFM $\boldsymbol{A}$. Our approach takes two steps: firstly, a polynomial reduction from a graph γ-colorability problem to hypergraph coloring problem, and then a polynomial reduction from hypergraph coloring problem to our problem of finding $\mathcal{S}_p$.

Before we start, we first introduce some related concepts and theorems. A graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is called γ-colorable if there exists a partition of $\mathcal{V}$ into γ sets, denoted by $\{\mathcal{V}_i\}_{i=1}^{\gamma}$, such that $|\mathcal{V}_i \cap \boldsymbol{e}_n| \leqslant 1$ for any $\boldsymbol{e}_n \in \mathcal{E}$. In other words, vertices in $\mathcal{G}$ can be colored using γ colors such that every pair of adjacent vertices have different colors. According to graph theory, it is NP-complete to decide whether a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is γ-colorable or not for any $\gamma \geqslant 3$.

A hypergraph $\mathcal{H}$ is defined by a set of vertices $\mathcal{V}$ and a set of hyperedges $\mathcal{E}$. Each hyperedge $\boldsymbol{e}_n \in \mathcal{E}$ can be incident to any number of vertices, i.e., $|\boldsymbol{e}_n| \geqslant 1$. $\mathcal{H}$ is called γ-uniform if every hyperedge is incident to γ vertices, i.e., $|\boldsymbol{e}_n| = \gamma$ for all $\boldsymbol{e}_n \in \mathcal{E}$. $\mathcal{H}$ is called strongly γ-colorable [43] if there exists a partition of $\mathcal{V}$ into γ sets, denoted by $\{\mathcal{V}_i\}_{i=1}^{\gamma}$, such that $|\mathcal{V}_i \cap \boldsymbol{e}_n| \leqslant 1$ for any $\boldsymbol{e}_n \in \mathcal{E}$. In other words, vertices of $\mathcal{H}$ can be colored using γ colors, such that every color appears at most once in every hyperedge.

We now prove that it is NP-complete to decide whether a γ-uniform hypergraph is strongly γ-colorable or not for any $\gamma \geqslant 3$.

Given any $\gamma \geqslant 3$ and a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with K vertices and N edges, we construct a hypergraph $\mathcal{H}(\mathcal{V}',\mathcal{E}')$ as follows: for every edge $\boldsymbol{e}_n \in \mathcal{E}$, we generate a hyperedge $\boldsymbol{e}'_n$ by adding to $\boldsymbol{e}_n$ $\gamma-2$ dummy vertices $\boldsymbol{v}_{n,1} \cdots \boldsymbol{v}_{n,\gamma-2}$. The resulted $\boldsymbol{e}'_n$ is incident to γ vertices, where $\gamma-2$ of them are dummy vertices only incident by $\boldsymbol{e}'_n$. Consequently, $\mathcal{H}$ is a γ-uniform hypergraph with $K+(\gamma-2)N$ vertices. Since $\boldsymbol{e}_n \subset \boldsymbol{e}'_n$ for all $n \in [1,N]$, $\mathcal{G}$ is a subgraph of $\mathcal{H}$. Then, the reduction is as follows:

- If $\mathcal{G}$ is γ-colorable, then for every hyperedge $\boldsymbol{e}'_n$, there are already two differently colored vertices. Then, by heuristically color the remaining $\gamma-2$ dummy vertices in $\boldsymbol{e}'_n$ using the remaining $\gamma-2$ colors for all $n \in [1,N]$, we obtain a strong $\gamma-$coloring of $\mathcal{H}$.
- If $\mathcal{H}$ is strongly γ-colorable, then since $\mathcal{G} \subset \mathcal{H}$, $\mathcal{G}$ is already colored with at most γ colors, say with $\gamma' \leqslant \gamma$ colors. To obtain a γ-coloring of $\mathcal{G}$, we can heuristically choose $\gamma-\gamma'$ vertices of $\mathcal{G}$ and assigning each of them with a different new color.

The above reduction indicates that an algorithm that can optimally solve the strong γ-coloring problem for any γ-uniform hypergraph will also solve the γ-coloring problem of any graph: given any $\mathcal{G}$, we construct $\mathcal{H}$ as described above and pass $\mathcal{H}$ to the algorithm for solution.

Then, by noting that it is NP-complete to decide whether $\mathcal{G}$ is γ-colorable or not, our reduction proves that it is NP-complete to decide whether a γ-uniform hypergraph is strongly γ-colorable or not.

We now prove the NP-completeness of deciding the existence of $\mathcal{S}_p$ by reducing a γ-uniform hypergraph to a SFM $\boldsymbol{A}$, and reducing the strong γ-coloring of $\mathcal{H}$ to $\mathcal{S}_p$.

Given any γ-uniform hypergraph $\mathcal{H}(\mathcal{V},\mathcal{E})$ with K vertices and N hyperedges, we construct an instance of $\boldsymbol{A}$ with K data packets and N receivers: we identify each $\boldsymbol{v}_k \in \mathcal{V}$ with a data packet $\mathbf{p}_k$, and identify each $\boldsymbol{e}_n \in \mathcal{E}$ with a receiver R_n who wants the data packets represented by the vertices in $\boldsymbol{e}_n$, i.e., letting $\mathcal{W}_n = \boldsymbol{e}_n$. In the resulting $\boldsymbol{A}$, every receiver wants γ data packets. Then, the reduction is as follows:

- If $\mathcal{H}$ is strongly γ-colorable, then since every $\boldsymbol{e}_n$ has γ vertices, we know that every color appears exactly once in every $\boldsymbol{e}_n$. In other words, there exists a partition $\{\mathcal{V}_i\}_{i=1}^{\gamma}$ of $\mathcal{V}$ such that $\mathcal{V}_i \cap \boldsymbol{e}_n = 1$ for all $n \in [1, n]$. Then, by identifying each $\mathcal{V}_i$ with a coding set $\mathcal{M}_i$, we have $|\mathcal{M}_i \cap \mathcal{W}_n| = 1$. Hence, $\mathcal{S} = \{\mathcal{M}_i\}_{i=1}^{\gamma}$ is a S-IDNC solution that allows all receivers to instantly decode one of their γ wanted data packets from the coded packet of each of the γ coding sets. According to Definition 3, $\mathcal{S}$ is a perfect S-IDNC solution $\mathcal{S}_p$.

- If $\boldsymbol{A}$ has the perfect S-IDNC solution $\mathcal{S}_p = \{\mathcal{M}_i\}_{i=1}^{\gamma}$, then $|\mathcal{M}_i \cap \mathcal{W}_n| = 1$ for all $n \in [1, N]$. Then by identifying each $\mathcal{M}_i$ with a vertex set $\mathcal{V}_i$, we obtain a partition $\{\mathcal{V}_i\}_{i=1}^{\gamma}$ of $\mathcal{V}$ such that $|\mathcal{V}_i \cap \boldsymbol{e}_n| = 1$ for all $n \in [1, N]$. By definition, this partition is a strong γ-coloring of $\mathcal{H}$.

The above reduction indicates that an algorithm that can optimally find the perfect solution for any SFM $\boldsymbol{A}$ will also solve the strong γ-coloring problem of any γ-uniform graph: given any such $\mathcal{H}$, we construct an instance of $\boldsymbol{A}$ as described above and pass $\boldsymbol{A}$ to the algorithm for solution.

Then, by noting that it is NP-complete to decide whether a γ-uniform hypergraph is strongly γ-colorable or not, our reduction proves that it is NP-complete to decide whether the perfect solution $\mathcal{S}_p$ exists or not for a given $\boldsymbol{A}$.

Appendix 2: Proof of Theorem 4

Theorem 4 requires the proof of $\chi(\overline{\mathcal{G}}_s) = \chi(\overline{\mathcal{G}}_g)$. Since every S-IDNC solution is also a G-IDNC solution, but a G-IDNC solution is not necessarily an S-IDNC solution, we have $U_s \geqslant U_g$, and thus $\chi(\overline{\mathcal{G}}_s) \geqslant \chi(\overline{\mathcal{G}}_g)$. Hence, here we only need to prove that $\chi(\overline{\mathcal{G}}_s) \leqslant \chi(\overline{\mathcal{G}}_g)$.

We first introduce the concept of *affiliated* S-IDNC graph $\mathcal{G}_{as}$ of a G-IDNC graph $\mathcal{G}_g$, which is construct as follows. Given $\mathcal{G}_g$ that involves K data packets and N receivers, we generate a graph $\mathcal{G}_{as}$ with K vertices, each representing a data packet. We then connect $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ in $\mathcal{G}_{as}$ if for every pair of $\{m, n\} \in [1, N]$, $\boldsymbol{v}_{i,m}$ and $\boldsymbol{v}_{j,n}$ are connected upon their existence in $\mathcal{G}_g$. In other words, we claim that $\mathbf{p}_i$ and $\mathbf{p}_j$ do not conflict if every vertex that represents $\mathbf{p}_i$ in $\mathcal{G}_g$ is connected to every vertex that represents $\mathbf{p}_j$ in $\mathcal{G}_g$.

Given an SFM $\boldsymbol{A}$, we can easily show that its S-IDNC graph $\mathcal{G}_s$ is the same as the affiliated S-IDNC graph $\mathcal{G}_{as}$ of its G-IDNC graph $\mathcal{G}_g$. Hence, our task becomes to prove that $\chi(\overline{\mathcal{G}}_{as}) \leqslant \chi(\overline{\mathcal{G}}_g)$, where $\chi(\overline{\mathcal{G}}_{as}) = U_s$. This statement is true if the following property is true:

After removing any clique $\mathcal{M}_g$ from $\mathcal{G}_g$, the chromatic number of the affiliated S-IDNC graph $\mathcal{G}_{as}$ is reduced by at most one.

Since $\mathcal{G}_g$ is nonempty as long as $\mathcal{G}_{as}$ is nonempty, this property indicates that any clique partition solution of $\mathcal{G}_g$ must have a size of at least $\chi(\overline{\mathcal{G}}_{as})$, which will prove that $\chi(\overline{\mathcal{G}}_{as}) \leqslant \chi(\overline{\mathcal{G}}_g)$. Property 6 can be proved through induction:

1. If $\mathcal{M}_g$ does not contain any conflicting data packets in $\mathcal{G}_{as}$, then $\chi(\overline{\mathcal{G}}_{as})$ is reduced by at most one;
2. If $\mathcal{M}_g$ contains one pair of conflicting data packets in $\mathcal{G}_{as}$, then $\chi(\overline{\mathcal{G}}_s)$ is reduced by at most one;
3. If $\mathcal{M}_g$ already contains m pairs of conflicting data packets in $\mathcal{G}_{as}$, then modifying $\mathcal{M}_g$ to contain one more pair of conflicting data packets in $\mathcal{G}_{as}$ cannot further reduce $\chi(\overline{\mathcal{G}}_{as})$.

The first statement is self-evident, because the set of data packets included in such $\mathcal{M}_g$ is a clique of $\mathcal{G}_{as}$. By removing it, $\chi(\overline{\mathcal{G}}_{as})$ can be reduced by at most one.

To prove the second statement, without loss of generality we assume that the pair of conflicting data packets is $(\mathbf{p}_1, \mathbf{p}_2)$. Then the set of data packets included in $\mathcal{M}_g$ takes a form of $\{\mathcal{M}_s, \mathbf{p}_1, \mathbf{p}_2\}$, where $\mathcal{M}_s$ is the set of pair-wise non-conflicting data packets, and thus is a clique of $\mathcal{G}_{as}$. Since $\mathbf{p}_1$ conflicts with $\mathbf{p}_2$, there exists at least one pair of unconnected vertices in $\mathcal{G}_g$ that represents $\mathbf{p}_1$ and $\mathbf{p}_2$. This pair is not included in $\mathcal{M}_g$, and thus is kept after removing $\mathcal{M}_g$ from $\mathcal{G}_g$. Hence, in the updated affiliated S-IDNC graph $\mathcal{G}'_{as}$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ exist, and are unconnected. Let the chromatic number of $\mathcal{G}'_{as}$ be U', then the minimum clique partition of $\mathcal{G}'_{as}$ takes a form of $\{\mathcal{M}_1, \cdots, \mathcal{M}_{U'}\}$, which keeps $\mathbf{p}_1$ and $\mathbf{p}_2$ in different coding sets. Then, since $\mathcal{M}_s$ is a clique of $\mathcal{G}_{as}$, $\{\mathcal{M}_s, \mathcal{M}_1, \cdots, \mathcal{M}_{U'}\}$ is a partition of $\mathcal{G}_{as}$ with a size of $U' + 1$. Thus, $U' \geqslant \chi(\overline{\mathcal{G}}_{as}) - 1$, implying that $\chi(\overline{\mathcal{G}}_s)$ is reduced by at most one after removing $\mathcal{M}_g$ from $\mathcal{G}_g$.

The proof of the third statement is similar to the second one, and thus is omitted here. According to the above three statements, no matter how many conflicting data packets are included in $\mathcal{M}_g$, after removing $\mathcal{M}_g$ from $\mathcal{G}_g$, the chromatic number of the affiliated S-IDNC graph $\mathcal{G}_{as}$ is reduced by at most one. Therefore, $\chi(\overline{\mathcal{G}}_g) \geqslant \chi(\overline{\mathcal{G}}_{as})$. Since $\mathcal{G}_{as}$ is the same as $\mathcal{G}_s$, we have $\chi(\overline{\mathcal{G}}_g) \geqslant \chi(\overline{\mathcal{G}}_s)$ and Theorem 3 is proved.

Competing interests

The authors declare that they have no competing interests.

Received: 29 June 2015 Accepted: 4 November 2015
Published online: 17 November 2015

References

1. JK Sundararajan, D Shah, M Médard, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*. ARQ for network coding, (2008)
2. R Ahlswede, N Cai, S Li, R Yeung, Network information flow. IEEE Trans. Inf. Theory. **46**, 1204–1216 (2000)
3. S Katti, H Rahul, W Hu, D Katabi, M Médard, J Crowcroft, XORs in the air: practical wireless network coding. IEEE/ACM Trans. Netw. **16**(3), 497–510 (2008)

4. C Fragouli, J Widmer, J Le Boudec, Efficient broadcasting using network coding. IEEE/ACM Trans. Netw. **16**(2), 450–463 (2008)
5. L Keller, E Drinea, C Fragouli, in *Proc.IEEE 4th Workshop on Network Coding, Theory, and Applications (NetCod)*. Online broadcasting with network coding, (2008)
6. A Eryilmaz, A Ozdaglar, M Médard, Ahmed E, On the delay and throughput gains of coding in unreliable networks. IEEE Trans. Inf. Theory. **54**(12), 5511–5524 (2008)
7. T Tran, T Nguyen, B Bose, in *Proc. IEEE 4th Workshop on Network Coding, Theory, and Applications (NetCod)*. A joint network-channel coding technique for single-hop wireless networks, (2008)
8. D Nguyen, T Tran, T Nguyen, B Bose, Wireless broadcast using network coding. IEEE Trans. Veh. Technol. **58**(2), 914–925 (2009)
9. DE Lucani, M Médard, M Stojanovic, in *Proc. IEEE 4th Workshop on Network Coding, Theory, and Applications (NetCod)*. Broadcasting in time-division duplexing: a random linear network coding approach, (2009), pp. 62–67
10. DE Lucani, M Médard, in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*. Random linear network coding for time-division duplexing: field size considerations, (2009)
11. J Heide, MV Pedersen, FHP Fitzek, T Larsen, in *Proc. IEEE Int. Conf. Communications (ICC) workshop*. Network coding for mobile devices—systematic binary random rateless codes, (2009)
12. S Sorour, S Valaee, in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*. Minimum broadcast decoding delay for generalized instantly decodable network coding, (2010)
13. E Rozner, AP Iyer, Y Mehta, L Qiu, M Jafry, in *Proc. ACM Int. Conf. Emerging Networking Experiments and Technologies (CoNEXT)*. ER: Efficient retransmission scheme for wireless LANs, (2007)
14. P Sadeghi, D Traskov, R Koetter, in *Proc. IEEE 5th Workshop on Network Coding, Theory, and Applications (NetCod)*. Adaptive network coding for broadcast channels, (2009), pp. 80–85
15. P Sadeghi, R Shams, D Traskov, An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels. EURASIP J. Wireless Commun. Netw. **2010**, 1–14 (2010)
16. P Sadeghi, M Yu, N Aboutorab, in *Proc. IEEE Int. Symp. Information Theory and its Applications (ISITA)*. On throughput-delay tradeoff of network coding for wireless communications, (2014), pp. 689–693
17. M Yu, P Sadeghi, N Aboutorab, From instantly decodable to random linear network coding. IEEE Trans Commun. **62**(11), 3943–3955 (2014)
18. T Ho, M Médard, R Koetter, D Karger, M Effros, J Shi, B Leong, A random linear network coding approach to multicast. IEEE Trans. Inf. Theory. **52**(10), 4413–4430 (2006)
19. J Barros, RA Costa, D Munaretto, J Widmer, in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. Effective delay control in online network coding, (2009), pp. 208–216
20. X Li, C-C Wang, Lin X, On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints. IEEE J. Sel. Areas Commun. **29**(5), 1094–1105 (2011)
21. L Yang, YE Sagduyu, JH Li, in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*. Adaptive network coding for scheduling real-time traffic with hard deadlines, (2012), pp. 105–114
22. SY El Rouayheb, MAR Chaudhry, A Sprintson, in *Proc. IEEE Information Theory Workshop (ITW)*. On the minimum number of transmissions in single-hop wireless coding networks, (2007)
23. S Sorour, S Valaee, in *Proc. IEEE. Int. Conf. Communications (ICC)*. On minimizing broadcast completion delay for instantly decodable network coding, (2010)
24. S Sorour, S Valaee, Coding opportunity densification strategies for instantly decodable network coding. IEEE Trans. Commun. **61**(12), 5077–5089 (2013)
25. N Aboutorab, P Sadeghi, S Sorour, Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems. IEEE Trans. Commun. **62**(4), 1269–1309 (2014)
26. N Aboutorab, P Sadeghi, SE Tajbakhsh, in *Proc. IEEE Int. Symp. Information Theory (ISIT)*. Instantly decodable network coding for delay reduction in cooperative data exchange systems, (2013), pp. 3095–3099
27. MS Karim, N Aboutorab, AA Nasir, P Sadeghi, in *Proc. IEEE Information Theory Workshop (ITW)*. Decoding delay reduction in network coded cooperative systems with intermittent status update, (2014), pp. 391–395
28. Y Keshtkarjahromi, H Seferoglu, R Ansari, A Khokhar, in *Proc IEEE Int. Conf. Computing, Networking and Communications (ICNC)*. Content-aware instantly decodable network coding over wireless networks, (2015), pp. 803–809
29. S Sorour, S Valaee, in *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Communications (PIMRC)*. Completion delay reduction in lossy feedback scenarios for instantly decodable network coding, (2011), pp. 2025–2029
30. S Sorour, S Valaee, in *Proc. IEEE ICC*. Completion delay minimization for instantly decodable network coding with limited feedback, (2011)
31. A Le, AS Tehrani, AG Dimakis, A Markopoulou, in *Proc. IEEE Int. Symp. Network Coding (NetCod)*. Instantly decodable network codes for real-time applications, (2013)
32. N Aboutorab, S Sorour, P Sadeghi, in *Proc. IEEE Int. Symp. Network Coding (NetCod)*. O2-gidnc: Beyond instantly decodable network coding, (2013), pp. 1–6
33. JM Harris, JL Hirst, MJ Mossinghoff, *Combinatorics and Graph Theory*, 2nd Edition. (Springer Press, Verlag New York, 2008)
34. M Kubale, *Graph Coloring, ser. Contemporary Mathematics*. (American Mathematical Society, Providence, Rhode Island, 2004)
35. WA Vasconcelos, CW Avery, WAJ Luxemburg, RE Shafer, D Marcus, DP Gelle, Advanced Problems: 5707-5713. American Mathematical Monthly. **77**, 84–85 (1970)
36. W Xiao, D Starobinski, in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. Extreme value FEC for reliable broadcasting in wireless networks, (2009)
37. M Ghaderi, D Towsley, J Kurose, in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. Reliability gain of network coding in lossy wireless networks, (2008)
38. B Swapna, A Eryilmaz, NB Shroff, Throughput-delay analysis of random linear network coding for wireless broadcasting. IEEE Trans Inf. Theory. **59**(10), 6328–6341 (2013)
39. B Ballobás, *Random Graphs*. (Cambridge University Press, Cambridge, UK, 2001)
40. B Bollobás, The chromatic number of random graphs. Combinatorica. **8**(1), 49–55 (1988)
41. R Costa, D Munaretto, J Widmer, J Barros, in *Proc. IEEE Int. Conf. Mobile Ad Hoc and Sensor Syst., (MASS)*. Informed network coding for minimum decoding delay, (2008), pp. 80–91
42. C Bron, J Kerbosch, Algorithm 457, Finding all cliques of an undirected graph. Commun. ACM. **16**(9), 575–577 (1973). New York
43. G Agnarsson, MM Halldórsson, in *Approximation and Online Algorithms*. Strong colorings of hypergraphs (Springer Berlin Heidelberg, 2005), pp. 253–266