A random graph theoretic framework for analyzing instantly decodable network codes

Shahriar Etemadi Tajbakhsh
Department of Engineering Science
University of Oxford
Oxford, UK
Email: shahriar.etemaditajbakhsh@eng.ox.ac.uk

Parastoo Sadeghi
Research School of Engineering
The Australian National University
Canberra, Australia
Email: parastoo.sadeghi@anu.edu.au

Justin Coon
Department of Engineering Science
University of Oxford
Oxford, UK
Email: justin.coon@eng.ox.ac.uk

***Abstract*—In this paper, we consider Instantly Decodable Network Codes (IDNC) in centralized and cooperative wireless networks. Analyzing the performance of these codes in terms of decoding delay under erasure channel assumption is highly complicated in general. We use some results from random graph theory and apply various relaxations to provide a closed form approximation of the average decoding delay. Our experiments show that the gap between analytical approximation and numerical results is small.**

I. Introduction

Carpooling, i.e. sharing a vehicle by multiple travellers, is a highly efficient way of utilizing the resources in transportation networks. Network coding, introduced in [1] plays a similar role in communication networks by combining multiple flows of information (or simply messages) using algebraic operations and with the goal of more efficient usage of network resources such as energy and bandwidth.

Network coding has been extensively studied in the literature over more than a decade and various techniques have been customized based on the requirements of different networks and applications. In addition to throughput and energy efficiency, delay is a key performance metric in real-time applications and live streaming.

There are two main families of network codes which perform differently in terms of delay. *Random linear network codes* (RLNC [2]), generate linear combinations of a set of messages with randomly selected coding coefficients from a sufficiently large finite field and forward them to the target receivers. The receivers need to collect sufficient number of independent linear combinations to be able to solve a system of equations and decode. RLNC is proven to be throughput optimal in multicast networks [2]. However, the receiver will not be able to obtain any information before receiving the entire set of transmitted packets which causes a considerable delay.

Unlike RLNC, instantly decodable network codes (IDNC) generate XOR combinations of the messages in a way that some (or all) of the receivers would be able to immediately decode the received packet and obtain a message. Therefore, IDNC is expected to be more efficient in terms of delay particularly if the number of receivers is not very large (possibly at the price of losing some throughput gain [3], [4]), and can be more convenient for delay sensitive applications.

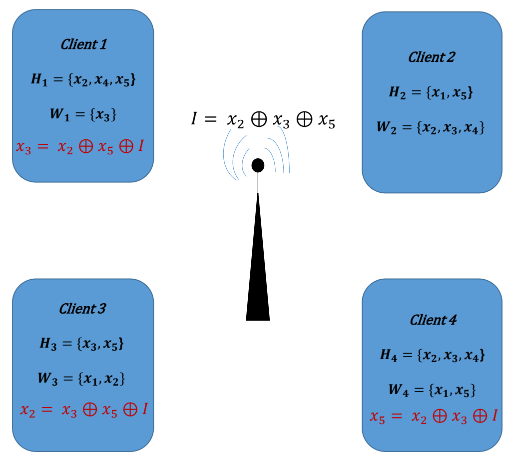

Fig. 1: An Instance of IDNC Problem

IDNC works based on the diversity in the side information available at receivers and demands of the receivers. As an example, suppose we have a group of four wireless users c_1, c_2, c_3, c_4 and a set of five messages $X = \{x_1, x_2, x_3, x_4, x_5\}$. We assume that each client c_i initially holds a subset $\mathcal{H}_i \subseteq X$ and is interested in receiving an arbitrary subset $\mathcal{W}_i \subseteq (X \backslash \mathcal{H}_i)$ which we refer to them as its *Has* set and *Wants* set, respectively. Suppose in this example, $\mathcal{H}_1 = \{x_2, x_4, x_5\}$, $\mathcal{H}_2 = \{x_1, x_5\}$, $\mathcal{H}_3 = \{x_3, x_5\}$, $\mathcal{H}_4 = \{x_2, x_3, x_4\}$ and $\mathcal{W}_1 = \{x_3\}$, $\mathcal{W}_2 = \{x_2, x_3, x_4\}$, $\mathcal{W}_3 = \{x_1, x_2\}$, $\mathcal{W}_4 = \{x_1, x_5\}$. In this example, the combination $I = x_2 \oplus x_3 \oplus x_5$ is instantly decodable for c_1, c_3 and c_4, where $\oplus$ denotes the XOR. For instance, c_3 is able to obtain message x_2 by calculating $x_2 = x_3 \oplus x_5 \oplus I$ (see Fig. 1). In general, an IDNC packet is an XOR combination of a set of messages where each target receiver holds all those messages within the combination except for one. Therefore, it XORs them with the received IDNC packet to obtain its desired message.

A. *Related Work and Our Contribution*

Our objective in this paper is to provide an analytical approximation of decoding delay performance of IDNC codes.

By mapping the IDNC problem to a graph theoretic model in [5], [6], finding an IDNC packet is translated to finding a maximum weighted clique which is well-known to be NP-hard [7]. Therefore, if all the target receivers decode and obtain their intended message successfully, the corresponding clique and all the edges connected to any vertex in the clique are removed from the graph. However, because of channel erasures some of these vertices would remain in the graph if their associated messages are not delivered successfully.

In this paper, we model the IDNC graphs with *random graphs* to analyze and provide a closed form approximation of the average decoding delay of such coded packets transmitted by a central base station over erasure channels (between the base station and each client). In its original form, formulating average decoding delay is a highly complex combinatoric problem in general. For instance, it is proven that for broadcast traffic where all the clients are interested in the entire set of messages, minimizing average decoding delay (although with a different definition of decoding delay from this paper) is an NP-hard problem [4]. Therefore, we apply different relaxations and approximations to the problem to be able to provide a closed form expression[1].

Our result relies on the well-known Bollobás-Erdös theorem [8] from random graph theory which gives the typical size of maximum clique in finite random graphs. We approximate the size of graph by a continuous function and use the mentioned theorem to obtain the rate of changes in the size of graph. This enables us to derive a closed form approximation for the *completion time* (the total number of required transmissions) and average *decoding delay* (to be formally defined later).

We also consider the problem of *cooperative index coding* [9] where the clients cooperate with each other by exchanging IDNC packets over a shared broadcast erasure channel instead of relying on a base station. Obtaining a similar approximation of the decoding delay in cooperative scenario is even further complicated. Therefore, in the cooperative scenario, we formulate the problem for a special case where the probability of erasure observed by all the clients is identical.

In this paper, we consider the algorithm in [6] as the base method for IDNC problem for our modelling purpose. However, we believe that our method can be tailored for modelling and formulating the performance of other variants of IDNC as well.

Several variants of IDNC in centralized scenarios have been proposed in the literature [6], [10], [11], [12], [13] to improve the performance of IDNC in terms of different metrics. Also delay reduction via IDNC in the cooperative setting has been studied in [14] and also in [15] for general network topologies. A special case of cooperative index coding problem where all the users are interested in the entire set of messages, known as *cooperative data exchange problem*, has been extensively studied in the literature [16], [17], [18]. Finding the minimum number of transmissions in this problem, as a special case of cooperative index coding, is formulated as a linear programming problem. However, the problem in general is NP-hard.

[1]It should be highlighted that although the approximation is neither a lower nor an upper bound, having a simple closed form expression is highly desirable and gives an insight about the impact of different parameters such as number of messages and receivers as well as the probability of erasure on decoding delay.

This paper is organized as follows. In Section II we establish our system model. We model the initial IDNC graph for both centralized and cooperative setting as a random graph in Section III. The process of updating the IDNC random graph by removing vertices during IDNC transmissions is formulated in Section IV. We compare our analytical expression for average decoding delay with numerical experiments in Section V and we conclude the paper in Section VI.

II. System Model

A group of n wireless users $C = \{c_1, \ldots, c_n\}$ and a set of k messages $X = \{x_1, \ldots, x_k\}$ are considered. Initially, a base station broadcasts all the messages in their original uncoded form to the set of users. Each user c_i receives each message x_ℓ with probability $1 - p_i$. Each user c_i is initially (before any transmission by the base station) interested in receiving an arbitrary subset of messages $\mathcal{R}_i \subseteq X$. We denote the cardinality of set $\mathcal{R}_i$ by $r_i = |\mathcal{R}_i|$. After the initial transmission of the entire set of messages by the base station, each user c_i might be missing a subset of messages $\mathcal{W}_i \subseteq \mathcal{R}_i$. Also at this stage, each user c_i initially holds a subset of messages $\mathcal{H}_i \subseteq X$. We refer to the sets $\mathcal{H}_i$ and $\mathcal{W}_i$ as the *Has* and *Wants* set of user c_i, respectively. This problem setting is called the *index coding problem* in the network coding literature [19].

We assume that after the initial transmissions of the messages, the transmitter (the base station in the centralized case) will have the full knowledge of the distribution of messages among the users which is obtained via error free acknowledgments (feedback) by the users. Then the transmitter at each round of transmission generates an *Instantly Decodable Network Coded (IDNC)* packet and broadcasts it to the users. An IDNC packet is an XOR combination of a subset of messages, i.e. $I = x_{i_1} \oplus \cdots \oplus x_{i_L}$ where each target receiver holds all the messages within this combination except for one message $x_{i_m}, 1 \leq m \leq L$. Therefore the target user is able to instantly decode and obtain x_{i_m} by subtracting the other messages within the combination. We assume the probabilities of erasure, p_i's, will remain constant over the initial transmissions period and recovery time.

We also discuss and formulate the cooperative variant of this problem for a special case in Section III-C. In this problem, we assume that after the initial transmissions by the base station the users will be cooperating with each other by exchanging IDNC packets rather than relying on the base station for recovering the missing messages.

In this paper we assume that the objective is to minimize the average decoding delay. After each transmission during the recovery phase, the decoding delay of any receiver who has not been able to decode a message is increased by one unit. The average decoding delay over all the users is calculated at the end.

III. Graph-based Model of IDNC

A. Deterministic Graph Model

In this section we briefly describe how the index coding problem is mapped to a graph based model based on the method developed in [5], [6]. The equivalent graph of an instance of index coding problem denoted by $G(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges, is formed as follows. Each message $x_\ell \in \mathcal{W}_i$ is associated with a vertex $v_{i\ell} \in \mathcal{V}$. There exist an edge $e_{i\ell jm} \in \mathcal{E}$ between any two vertices $v_{i\ell}$ and v_{jm} if one of the following conditions holds.

- condition 1: $\ell = m$.
- condition 2: $x_m \in \mathcal{H}_i$ and $x_\ell \in \mathcal{H}_j$.

In other words, there exists an edge between two vertices either if the two users are missing the same messages or if each of them holds the wanted message by the other one. It is easy to see that x_ℓ (if condition 1 holds) or the combination $x_\ell \oplus x_m$ (where the condition 2 holds) is instantly decodable for both c_i and c_j. Therefore, any clique in such a graph is equivalent to an instantly decodable packet for all the target receivers. If the probabilities of erasure is identical for all the channels between the base station and the users, finding the maximum clique is expected to minimize the average decoding delay. If an IDNC packet is successfully received by a receiver, the corresponding vertex is removed. Otherwise the vertex will remain in the graph. If the probabilities of erasure are different for various receivers, choosing the maximum clique does not necessarily result in removing the largest number of vertices as the vertices associated with the receivers with higher probability of erasure are less likely to be removed. Therefore, in [6], a heuristic weighting mechanism for the vertices is proposed to find the clique which results in removing more vertices. At each round, the algorithm attempts to find the *maximum weighted clique* and transmits the equivalent packet. An updated IDNC graph is generated after receiving the acknowledgements by the users. The algorithm continues until all the vertices are removed. It has to be re-emphasized that finding the maximum clique (and consequently minimizing the decoding delay) is an NP hard problem in general.

B. Random Graph Model

Since the erasures overs wireless channels (and consequently the formation of the Wants sets of the clients) are random by nature, it is reasonable to translate the IDNC graph model to a random graph model to facilitate the analysis of the problem by exploiting some well-known tools from random graph theory. In this subsection we describe how the IDNC problems are mapped to random graphs according to the probabilities of erasure and number of required messages by each user. A random graph $\mathcal{G}(N,\pi)$ is identified by the number of vertices in the graph N (the size of graph) and the probability of existing an edge between any two vertices in the graph π. Since the size of the random graph varies after each transmission, we denote the IDNC random graph by $\mathcal{G}(N(t),\pi(t))$ where $t = 0,1,\ldots,T$ is the discrete index of time and T is the total number of required transmissions to recover the entire set of wanted messages.

As mentioned earlier, in this paper we apply various mathematical relaxations and heuristic simplifications to maintain the problem tractable. As one of these approximations, we replace any Bernoulli random variable with its expectation unless otherwise mentioned. Therefore, we assume that the number of missing messages by user c_i is $r_i p_i$ immediately after the initial transmissions. Therefore the initial size of graph is assumed to be $N(0) = \sum_{i=1}^{n} r_i p_i$.

The initial probability of an edge existing between any two vertices in $\mathcal{G}$ is calculated as follows. We assume that:

$$\pi(0) = \frac{E\{|\mathcal{E}|\}}{\frac{N(0)(N(0)-1)}{2}} \tag{1}$$

where $E\{|\mathcal{E}|\}$ is the expected number of edges in the graph and $\frac{N(0)(N(0)-1)}{2}$ is the total number of pairs of vertices in the graph. We denote the subset of edges representing one message in the Wants set of c_i and the other one in the Wants set of c_j by $\mathcal{E}_{ij} = \{(v_{i\ell}, v_{jm}) : e_{i\ell jm} \in \mathcal{E}\}$. Therefore we have,

$$E\{|\mathcal{E}|\} = \sum_{i=1}^{n}\sum_{j=i+1}^{n} E\{|\mathcal{E}_{ij}|\} \tag{2}$$

We further split the subset $\mathcal{E}_{ij}$ to two subsets $\mathcal{E}_{ij}^s$ and $\mathcal{E}_{ij}^d$ which are the subsets corresponding to edges generated because of condition 1 and condition 2, respectively (the previously mentioned two conditions of an edge existing between two vertices). In other words $\mathcal{E}_{ij}^s = \{e_{i\ell jm} : \ell = m\}$ and $\mathcal{E}_{ij}^d = \{e_{i\ell jm} : \ell \neq m\}$. Hence, $\mathcal{E}_{ij} = \mathcal{E}_{ij}^s \cup \mathcal{E}_{ij}^d$.

$$\begin{aligned} E\{|\mathcal{E}_{ij}|\} = E\{|\mathcal{E}_{ij}^s|\} + E\{|\mathcal{E}_{ij}^d|\} = \sum_{a=\max\{|\mathcal{W}_i|+|\mathcal{W}_j|-k,0\}}^{\min\{|\mathcal{W}_i|,|\mathcal{W}_j|\}} aP\{|\mathcal{E}_{ij}^s| = a\} + \\ \sum_{a=\max\{|\mathcal{W}_i|+|\mathcal{W}_j|-k,0\}}^{\min\{|\mathcal{W}_i|,|\mathcal{W}_j|\}} P\{|\mathcal{E}_{ij}^s| = a\}(1-p_i)(1-p_j)(|\mathcal{W}_i|-a)(|\mathcal{W}_j|-a) \end{aligned} \tag{3}$$

where

$$P\{|\mathcal{E}_{ij}^s| = a\} = \frac{\binom{k}{a}\binom{k-a}{\min\{\lfloor r_i p_i\rfloor, \lfloor r_j p_j\rfloor - a\}}\binom{k-\min\{\lfloor r_i p_i\rfloor, \lfloor r_j p_j\rfloor\}}{\max\{\lfloor r_i p_i\rfloor, \lfloor r_j p_j\rfloor - a\}}}{\binom{k}{\lfloor r_i p_i\rfloor}\binom{k}{\lfloor r_j p_j\rfloor}} \tag{4}$$

It should be noted that the r.h.s of Equation (3) consists of two terms where the first term represents the expected size of $\mathcal{E}_{ij}^s$ (created by the common messages missing by both c_i and c_j) and the second term obtains the expected number of edges between the rest of the vertices associated with uncommon messages (given that a messages are missing in common by both of the clients). Also the numerator in equation (4) calculates the total number of possible positions for *Wanted* messages by c_i and c_j given that they both want the same a messages plus some other messages. The lower and upper limits of the two summations in equation (3) indicates the minimum and maximum possible number of common messages in the Wants sets of both c_i and c_j, i.e. $\mathcal{W}_i \cap \mathcal{W}_j$. If $|\mathcal{W}_i| + |\mathcal{W}_j| \leq k$, it is possible that there is no overlap between the wants sets of two messages and if $|\mathcal{W}_i|+|\mathcal{W}_j| > k$ there are at least $|\mathcal{W}_i| + |\mathcal{W}_j| - k$ common messages in the Wants sets of both c_i and c_j. Obviously the number of common messages cannot exceed the minimum of the size of the two sets $\mathcal{W}_i$ and $\mathcal{W}_j$. As mentioned earlier the binomial random variables have been replaced by their expectation. Since these expected values might not be integers, we have approximated them by their floor value, e.g. $\lfloor r_i p_i \rfloor$. Therefore using a straightforward combinatoric argument the numerator is obtained: The a common messages can take $\binom{k}{a}$ different placements. The $\min\{\lfloor r_i p_i \rfloor, \lfloor r_j p_j \rfloor\} - a$ uncommon messages in the smaller Wants set (between c_i and c_j) can take any of the $k - a$ available places and the $\max\{\lfloor r_i p_i \rfloor, \lfloor r_j p_j \rfloor\} - a$ uncommon messages in the larger Wants set can take the remaining $k - \min\{\lfloor r_i p_i \rfloor, \lfloor r_j p_j \rfloor\}$ places. Also the denominator is the total number of possibilities of placing $\lfloor r_j p_j \rfloor$ and $\lfloor r_j p_j \rfloor$ wanted messages each in k places. Therefore, the entire fraction provides the probability of having a wanted messages in common, between c_i and c_j. It should be noted that in the derivation of $\pi(0)$ we have assumed that only the expected number of required messages r_i is given without any information about the position of these messages.

Another key parameter essential in understanding how the random graph is updated is the expected probability of erasure. As mentioned earlier, because of the erasure over the wireless links between the base station and the clients, not all of vertices in the discovered maximum weighted clique would necessarily be removed from the IDNC graph as the IDNC packet may not be successfully received at some target recipients. Finding how likely it is that a specific client c_i is among the target recipients of an IDNC packet is extremely complex. Therefore, as a natural approximation we assume that the probability of a specific client c_i being targeted by the IDNC packet is proportional to its cardinality of Wants set $|\mathcal{W}_i|$ (in other words, we assume all the vertices in an IDNC graph have equal chance to be included in the clique). Hence, the overall probability of missing an IDNC packet, denoted by p_e is obtained as follows.

$$p_e = \sum_{i=1}^{n} \Big(\frac{r_i p_i}{\sum_{j=1}^{n} r_j p_j} \Big) p_i = \frac{\sum_{i=1}^{n} r_i p_i^2}{\sum_{j=1}^{n} r_j p_j} \tag{5}$$

C. Cooperative Index Coding Problem

A closely related problem to the index coding problem is introduced in [9]. In this problem, instead of relying on the base station for recovering the wanted messages, the users exchange IDNC packets until they all obtain their wanted messages. In this problem, in addition to the main IDNC graph, for each user c_l a *local graph* $G_l(\mathcal{V}_l, \mathcal{E}_l)$ is formed [14]. There exists a vertex $v^l_{i\ell} \in \mathcal{V}_l$ for each $x_\ell \in \mathcal{W}_i \cap \mathcal{H}_l$. There exists an edge $e^l_{i\ell jm} \in \mathcal{E}_l$ between two vertices $v^l_{i\ell}$ and v^l_{jm} if one of the two following conditions hold.

- condition 1: $\ell = m$.
- condition 2: $x_m \in \mathcal{H}_i$ and $x_\ell \in \mathcal{H}_j$.

In other words, $\mathcal{V}_l \subseteq \mathcal{V}$ and $\mathcal{E}_l \subseteq \mathcal{E}$. Similar to the centralized case, a clique in any local graph G_l is equivalent to an IDNC packet which the user c_l is able to generate. Minimizing the decoding delay is obviously NP-hard as the problem is reduced to the centralized IDNC problem if one of the clients holds the entire set of messages. As a heuristic solution to this problem, in this paper we assume that at each round, all the clients assign weight to their local graphs according to the weighting mechanism in [6] and find the maximum weight clique in their corresponding local graph. Among the set of these maximum cliques (each from one local graph), the clique with maximum weight is selected (i.e. maximum of maximum weighted cliques) and the corresponding IDNC packet is transmitted until all the wanted messages by all the clients are delivered. The main graph and the local graphs are updated accordingly.

Modeling the cooperative problem as a random graph is more complicated than the centralized version. This is mainly because the probability of erasure between any two clients and also the probability of existing an edge between vertices are different for each local graph. Therefore, the size of selected clique at each round and the probability of successful delivery of messages will be depending on which local graph is chosen to provide the clique (or equivalently which client is selected to transmit the IDNC packet). In this paper, we consider a simpler case where all the probabilities of erasure between the base station and each user are identical (denoted by p) and the probability of erasure between any pair of clients is identically q. Therefore the expected number of vertices in the random graph is $N(0) = rp$ at $t = 0$, where $r = \sum_{i=1}^{n} r_i$. To derive the *effective* probability of existing an edge between any two vertices in the graph we should take the fact into account that only the vertices belonging to any given local graph can be removed at the same round of transmission. Therefore, the probability of existence of an edge between any two vertices $v_{i\ell}$ and v_{jm} in the main graph drawn from the subset of vertices V_l , denoted by π^c is obtained as follows.

$$\begin{aligned}
&\pi^c(0) = P(v_{jm} \in V_l) P(i \neq l, j \neq l, i \neq j) \ldots \\
&\ldots P(\ell \neq m) P(x_m \in \mathcal{H}_i) P(x_\ell \in \mathcal{H}_j) \ldots \\
&\quad \cdots + P(v_{jm} \in V_l) P(i \neq l, j \neq l, i \neq j) P(\ell = m) = \\
&\qquad (1-p)\frac{n-2}{n}\Big[\frac{k-1}{k}(1-p)^2 + \frac{1}{k}\Big]
\end{aligned} \tag{6}$$

In other words, a vertex $v_{i\ell}$ is taken and the probability that the vertex v_{jm} belongs to the same local graph and corresponds to a different client $i \neq j$ and one of the two conditions holds is obtained. Also, it should be noted that for this scenario $p_e = q$. In the next section we will discuss how the random graph model can be exploited to estimate the average decoding delay.

IV. Clique Removal Process

At each round of transmissions a subset of vertices and all their corresponding edges (edges with at least one vertex in the subset) are removed from the IDNC graph. Interestingly, there is a possibility of formation of new edges in the graph between the remaining vertices after each transmission (we will discuss the details shortly). In general finding the number of edges by removing a subset of vertices (belonging to the discovered clique) is a highly complex graph theoretic problem. It should be noted that this is different from graph decomposition problem where only the edges of the graph are partitioned rather than vertices and edges together. In this paper, we apply two main heuristic relaxations to the problem to provide an approximation of the decoding delay which are discussed (mostly qualitatively) in the following.

- *Relaxation 1*: *We assume that the vertices in the clique are drawn i.i.d from the set of remaining vertices at each round of transmission and $\pi(t)$ remains constant.*
  The distribution of vertices' degrees is expected to be binomial in an Erdös-Rényi random graph. However, the distribution of the vertices forming the maximum weighted clique is unknown and hard to find. Therefore, in the absence of any information about such a distribution one can naturally assume that the vertices are drawn i.i.d from the set of vertices. If we accept this assumption, since the existence of each edge between any two vertices is independent of others, it is easy to show that $\pi(t)$ is expected to remain constant. However, there are two factors which are not considered in this assumption. Firstly, vertices with higher degrees are more likely to be included in the clique. Secondly, it is interesting to mention that in special cases it is possible that a new edge is created in the graph after a transmission. This occurs when $x_\ell \in \mathcal{W}_i \cup \mathcal{W}_j$, $x_m \in \mathcal{W}_i$ and $x_m \in \mathcal{H}_j$. In this case, if x_ℓ is delivered to c_j but not to c_i, the edge $e_{i\ell j\ell}$ is removed. However the new edge $e_{i\ell jm}$ is added to the graph. In other words these two factors act opposite to each other. Therefore we ignore the effect of the both factors and as a heuristic approximation we assume that the vertices are drawn i.i.d and hence, $\pi(t) = \pi(0)$.
- *Relaxation 2*: *We assume that the maximum clique is selected for removal from the graph at each round of transmission rather than the maximum weighted clique.*
  We will be using the well known theorem in [8] to identify the rate of changes in the graph later in this section. Therefore we need to assume that we select the maximum clique at each round to be able to use the theorem. However, it should be noted that the main purpose of weighting is to include vertices with smaller probability of erasure (hence more likelihood of successful delivery) with higher priority. If we denote the size of maximum clique and maximum weighted clique at round t by χ_t and χ_t^w, respectively, it is obvious that $\chi_t^w \leq \chi_t$. Therefore, while the selected vertices are more likely to be delivered successfully, however less number of vertices are included in the maximum weighted clique. Hence, it is reasonable to assume that choosing the maximum clique would have a similar impact on the average reduction in decoding delay.

We approximate the functions $N(t)$ and $\pi(t)$ with a continuous function of time t. Using the well-known Bollobás-Erdös theorem we know that in a finite graph the size of maximum clique can be approximated by $\frac{2\log N}{log\frac{1}{\pi}}$ where N is the size of the random graph and π is the probability of existing an edge. Therefore we can derive the rate of changes in the size of IDNC random graph given the above mentioned relaxations.

$$\frac{dN(t)}{dt} = -2(1-p_e)\frac{\log N(t)}{\log\frac{1}{\pi(0)}} \tag{7}$$

We need to find the expected total number of required transmissions T. Therefore,

$$\frac{dt}{dN(t)} = -\frac{\log\frac{1}{\pi(0)}}{2(1-p_e)\log N(t)} \tag{8}$$

Hence, we have:

$$T' = -\int_{N(0)}^{\epsilon} \frac{\log\frac{1}{\pi(0)}}{2(1-p_e)\log N} dN = \frac{\log\frac{1}{\pi(0)}}{2(1-p_e)} \operatorname{li}(N)\Big|_{N(0)}^{\epsilon} \tag{9}$$

where $1 < \epsilon \ll N(0)$ and $\operatorname{li}(\cdot)$ is the logarithmic integral function (series are available to obtain its value numerically). p_e, $N(0)$ and $\pi(0)$ were obtained for centralized and cooperative scenarios in Section III. T' is the number of transmissions which reduces the size of graph from $N(0)$ to ϵ. To have a better approximation we assume the last ϵ remaining vertices are isolated and are removed individually. Therefore,

$$T = T' + \frac{\epsilon}{1-p_e} \tag{10}$$

From the definition of decoding delay, we know that for each user who is not able to decode a message at each round of transmission the decoding delay is increased by one unit. Therefore at each round of transmission $n - \frac{dN}{dt}$ units is added to the total decoding delay observed by all the users. Therefore, if we denote the average decoding delay per user by D, in the continuous domain it is approximated as follows.

$$D = \frac{1}{n}\int_0^T \left(n - \frac{dN}{dt}\right)dt = \frac{1}{n}(nT - N) \tag{11}$$

Hence, a closed form approximation of the average decoding delay is obtained.

V. Numerical Experiments

In this section we compare the analytical approximation of average decoding delay provided in Section IV with the results of numerical experiments both for the centralized and cooperative scenario. In the first set of experiments, we have considered four different initial settings with different values of n, k, p_i's and R_i's. The details of these parameters are represented in Table I, where p_i's and R_i's are selected randomly

	$n{=}20, k{=}20$, $10 \le R_i \le 15$ $0.3 \le p \le 0.5$	$n{=}15, k{=}30$, $15 \le R_i \le 25$ $0.2 \le p \le 0.4$	$n{=}10, k{=}20$, $15 \le R_i \le 20$ $0.1 \le p \le 0.5$	$n{=}10, k{=}10$, $8 \le R_i \le 10$ $0.2 \le p \le 0.6$
D_{sim}	35.90	28.11	14.34	16.42
D_{theory}	35.54	25.32	13.07	15.97

TABLE I: average decoding delay: analysis vs. simulations

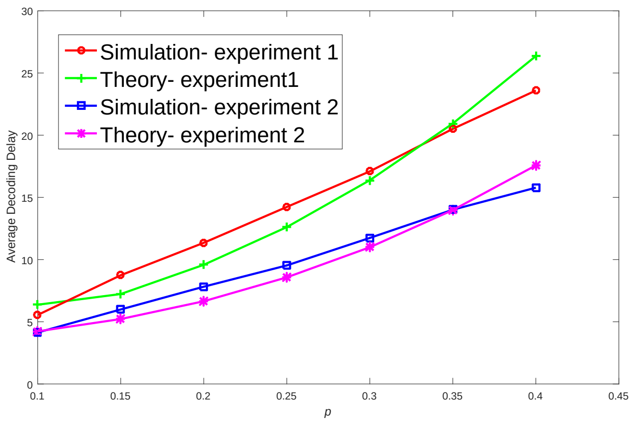

Fig. 2: Average decoding delay in cooperative scenario

within the given range. For each setting, we have repeated the experiment for 1000 times and averaged the mean decoding delay over these iterations. D_{sim} and D_{theory} are the average decoding delay from simulations and analysis, respectively. We have fixed $\epsilon = 7$ for all the experiments. In the second set of experiments we have considered the cooperative index coding problem, where the average decoding delay has been obtained via numerical experiments for $0.1 \leq p \leq 0.4$ (with steps of 0.05) for two different initial settings. For each value of p, we have averaged the decoding delay over 500 iterations. In the first setting we have set $n = 20$, $k = 20$, $q = 0.3$, $\hat{R} = 200$ and $\epsilon = 3$ (where $\hat{R}$ is the total number of required messages). In the second experiment, we have changed all these parameters except for ϵ where $n = 10$, $k = 25$, $\hat{R} = 150$ and $q = 0.2$. The results are compared with the analytical approximation in Fig. 2, where the first and second initial setting are labeled as experiment 1 and 2, respectively. As it can be observed despite the various approximations we had applied in Section III and IV, the analytical and numerical results are reasonably close.

VI. Conclusion

In this paper, we provided an analytical approximation method for average decoding delay of instantly decodable network codes for the index coding problem by mapping the IDNC problem to a random graph theoretic model. We considered both the centralized and cooperative settings. We compared our analytical result with numerical experiences which are desirably close. We believe that the method developed in this paper can be a basis for analyzing and understanding the delay performance of more complicated network coding algorithms particularly IDNC problems, by applying appropriate modifications to our model in the future.

Acknowledgment

This work was supported in part by EPSRC grant number EP/N002350/1 (Spatially Embedded Networks).

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. Yeung, "Nework information flow," *IEEE Transaction on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.

[3] M. Yu, P. Sadeghi, and N. Aboutorab, "Performance characterization and transmission schemes for instantly decodable network coding in wireless broadcast," *EURASIP Journal on Advances in Signal Processing*, no. 1, pp. 1–17, 2015.

[4] M. Yu, A. Sprintson, and P. Sadeghi, "On minimizing the average packet decoding delay in wireless network coded broadcast," in *International Symposium on Network Coding (NetCod)*, 2015.

[5] M. A. R. Chaudhry and A. Sprinston, "Efficient algorithms for index coding," in *Proc. IEEE Infocom Workshops*, 2008.

[6] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[7] M. Garey and D. Johnson, "Computers and intractability-a guide to the theory of np-completeness," 1979.

[8] B. Bollobás and P. Erdös, "Cliques in random graphs," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 80, pp. 419–427, 1976.

[9] S. E. Tajbakhsh and P. Sadeghi, "Coded cooperative data exchange for multiple unicasts," in *Proc. Information Theory Workshop (ITW 2012)*, Laussane, Swizerland, September 2012.

[10] S. Sorour, A. Douik, S. Valaee, T. Al-Naffouri, and M. S. Alouini, "Partially blind instantly decodable network codes for lossy feedback environment," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 4871–4883, 2014.

[11] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Completion time reduction in instantly decodable network coding through decoding delay control," in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 5008–5013.

[12] S. Sorour, N. Aboutorab, P. Sadeghi, M. S. Karim, T. Y. Al-Naffouri, and M. Alouini, "Delay reduction in persistent erasure channels for generalized instantly decodable network coding," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.

[13] S. Sorour and S. Valaee, "On densifying coding opportunities in instantly decodable network coding graphs," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2012, pp. 2456–2460.

[14] N. Aboutorab, P. Sadeghi, and S. E. Tajbakhsh, "Instantly decodable network coding for delay reduction in cooperative data exchange systems," in *Proc. IEEE International Symposium on Information Theory (ISIT), 2013*. IEEE, 2013.

[15] S. E. Tajbakhsh, P. Sadeghi, and N. Aboutorab, "Instantly decodable network codes for cooperative index coding problem over general topologies," in *Australian Communications Theory Workshop (AusCTW)*, 2014.

[16] S. E. Rouayheb, A. Sprinston, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. IEEE Information Theory Workshop (ITW)*, Cairo, Egypt, Jan. 2010, pp. 118–122.

[17] A. Sprinston, P. Sadeghi, G. Booker, and S. E. Rouayheb, "A randomized algorithm and performance bounds for coded cooeprative data exchange," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 1888–1892.

[18] S. E. Tajbakhsh, P. Sadeghi, and R. Shams, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *Proc. The 2011 International Symposium on Network Coding (Netcod)*, Beijing, China, July 2011.

[19] Y. Birk and T. Col, "Coding-on-demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients," *IEEE Transaction on Information Theory*, vol. 52, pp. 2825–2830, 2006.