Enabling a Tradeoff between Completion Time and Decoding Delay in Instantly Decodable Network Coded Systems

Neda Aboutorab, *Member, IEEE*, Parastoo Sadeghi, *Senior Member, IEEE*, and Sameh Sorour, *Member, IEEE*

Abstract—This paper studies the complicated interplay of the completion time (as a measure of throughput) and the decoding delay performance in instantly decodable network coded (IDNC) systems over wireless broadcast erasure channels with memory. We propose two new algorithms that enable a tradeoff for an improved balance between completion time and decoding delay of broadcasting a block of packets. We first formulate the IDNC packet selection problem that improves the balance between completion time and decoding delay as a statistical shortest path (SSP) problem. However, since finding such packet selection policy using the SSP technique is computationally complex, we employ its geometric structure to find some guidelines and use them to propose two efficient heuristic packet selection algorithms for broadcast erasure channels with a wide range of memory conditions. It is shown that each one of the two proposed algorithms is superior for a specific range of memory conditions. Furthermore, we show that the proposed algorithms achieve an improved fairness in terms of the decoding delay across all receivers.

Index Terms—Instantly Decodable Network Coding, Decoding delay, Completion time, Broadcast, Gilbert-Elliott channels.

I. INTRODUCTION

NETWORK coding (NC) [1]–[3] generally refers to mixing different information flows at the sender or intermediate nodes in a data communication network. It has been shown that NC can substantially improve the throughput of many wireless communication systems [2], [3]. As a result, it has become a promising candidate for delivering high data rate content in future wireless communication networks. However, there are high data rate applications that also require low delay. It is known that the higher throughput offered by NC does not necessarily translate into faster delivery of information to the application [4], [5]. In general, the mixed information needs to be disentangled or network decoded first. Understanding the interplay between throughput and delay and devising NC schemes that strike a balance between the two are particularly important, which has proven to be challenging [4]–[19].

Manuscript received March 5, 2013; revised September 2 and November 25, 2013. The editor coordinating the review of this paper and approving it for publication was T. Ho.

N. Aboutorab and P. Sadeghi are with the Research School of Information Sciences and Engineering, Australian National University, Canberra, ACT 0200, Australia (e-mail: {neda.aboutorab, parastoo.sadeghi}@anu.edu.au). Their work was supported under Australian Research Councils Discovery Projects funding scheme (project no. DP120100160).

S. Sorour is with the Electrical Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Eastern Province, Saudi Arabia (e-mail: samehsorour@kfupm.edu.sa).

Digital Object Identifier 10.1109/TCOMM.2014.021614.130172

An important example that illustrates such tension between throughput and delay is random linear network coding (RLNC) [6], [16], [18], [20] in broadcast erasure channels. In RLNC, the sender combines a frame or block of N packets using random coefficients from a finite field and broadcasts different combinations until all receivers have received N linearly independent coded packets. In this case, RLNC achieves the best throughput (measured in block completion time) [4], [6], [16]. However, the delay performance may not be desirable, as decoding at the receivers is generally only possible after N independent coded packets are successfully received.

In order to reduce the decoding delay in broadcast erasure channels using NC, an attractive strategy is to employ instantly decodable NC (IDNC). As the name suggests, IDNC aims to provide instant packet decoding at the receivers upon successful packet reception, a property that RLNC does not guarantee. The following tension between throughput and delay is observed in IDNC. Compared to RLNC, IDNC incurs a generally higher completion time for the broadcast of the same number of N packets. Nevertheless, it can provide earlier packet decoding, which can be beneficial and result in lower delays, especially in applications in which every decoded packet brings new information to the destination irrespective of its order. Examples include multiple-description source coded systems [21], as well as broadcasting a set of equally important emergency messages, safety messages or map updates to a group of wireless receivers, sensors, or vehicles. In such applications, joint control over completion time and decoding delay can be particularly important, where the transmission of a block of packets should be completed as fast as possible, and at the same time fast decoding of packets is necessary.

Inspired by the low-complexity XOR-based encoding and decoding process of IDNC and its potential in the aforementioned applications, as well in multimedia broadcast or multicast services (MBMS) and even in unicast settings [5], [13]–[15], [17], [22]–[25], [23], [24], [26]–[29], in this paper we are interested in understanding the interplay between its throughput and delay over broadcast erasure channels.

Two main classes of IDNC schemes, namely strict IDNC (SIDNC) and generalized IDNC (GIDNC), are broadly embraced in the literature [5], [13]–[15], [27], [29]. SIDNC, considered in [5], [13], [27], forces the sender to transmit coded packets including *at most* one missing packet for each

of the receivers. This strict requirement of SIDNC can limit its coding opportunities. GIDNC [14], [15], [29] relaxes this requirement and allows packet combinations that can target more receivers with an instantly decodable packet, but may contain more than one new packet for some other receivers, which they will discard. It is shown that GIDNC can result in better throughput [15], [25] or decoding delay performance [14] compared to SIDNC. Therefore in this paper, we have adopted GIDNC as our IDNC scheme, and for the rest of the paper refer to it as IDNC for short.

The problem of throughput optimization in IDNC systems is investigated in [24], where in each transmission the sender forms a new packet by XORing a maximum set of the lost packets from different receivers, and thus addresses the maximum set of receivers by an IDNC packet. However, in [24] other performance metrics such as delay are not considered. In another work, authors in [30] consider stochastic arrivals of packets and a sliding window coding approach, where every transmission is a linear combination of packets from the window. Coding is considered to be performed over finite field $\mathbb{F}_q$, and the goal is to reduce the queue size and delay while achieving 100% throughput optimality. The problem of maximizing the throughput for a deadline-constrained video-streaming scenario is studied in [23], [26], where an infinite backlog of packets is considered at the sender. Nonetheless, the throughput analysis is provided for only the two- and three-receiver cases. Furthermore, it is assumed each packet has a delivery deadline and has to be decoded before the deadline, otherwise it is expired. In this paper, however, we consider the transmission of a finite-length block of packets, where all the packets in the block have to be received by all the receivers and there is no explicit packet deadline. In addition, in this paper, no new packet arrival is considered in the system while the transmission of a block is in progress. We also consider a general number of receivers.

In this context, the closest works to ours are [14], [25] and [15]. In particular, the authors in [14] aimed to improve the decoding delay of a generalized IDNC scheme. They showed that for a lower decoding delay, maximum number of receivers with the lowest packet erasure probabilities should be targeted in each IDNC transmission. In separate works [15], [25], the same authors aimed to improve the completion time of IDNC. They showed that for this purpose, the receivers with the maximum number of missing packets with the highest erasure probabilities should be targeted in each IDNC transmission.

A close study of [14], [15], [25] reveals that trying to improve either IDNC's decoding delay or completion time on its own can result in undermining the other performance metric. In other words, while trying to improve the decoding delay, the receiver(s) with the maximum number of missing packets may remain untargeted, which can increase the completion time. Also trying to improve the completion time may limit the total number of receivers that can be targeted in each IDNC transmission, which can increase the decoding delay. To the best of our knowledge, there is no joint control of completion time and decoding delay for IDNC schemes in the literature. Thus, in this paper, our objective is to take a holistic approach, in which the completion time and decoding delay of IDNC are taken into account at the same time.

In addition, we have observed that the decoding delay across various receivers in IDNC schemes of [14], [25] and [15] can vary significantly. This may not be desirable in MBMS or other applications which should guarantee a certain quality of service across all receivers. These observations lead us to the following open problems:

Is there an IDNC scheme that can offer a balanced performance in terms of completion time and decoding delay and can also provide a more uniform or fair decoding delay across all receivers for the broadcast of N packets in erasure channels?

To address these questions in this paper, we propose a new IDNC transmission scheme which builds upon the contributions in [14], [25] and [15]. At its core, our proposed scheme recognizes that 1) the completion time of each individual receiver is determined not only by the number of packets it is missing, but also by the number of IDNC transmissions in which it is not targeted (while still needing a packet(s)) and 2) the overall IDNC completion time is the maximum of individual completion times. Therefore, our IDNC transmission scheme gives priority to the receivers that have the highest expected completion time so far. More precisely, the priority of each receiver is the sum of two terms: The first term is its number of missing packets divided by its average packet reception probability. This is the expected number of transmissions to serve this receiver if it is targeted in all following transmissions. The second term is the decoding delay the receiver has experienced so far. Under this scheme, a receiver with a small number of missing packets which has remained untargeted in a number of previous transmissions may take precedence over other receivers. Hence, our scheme tends to equalize the decoding delay experience across the receivers. Furthermore, we will extend our proposed scheme to the case of broadcast erasure channels with memory [31], where the packet erasures occur in bursts, due to deep fading and shadowing. By following the proposed channel models in [13], [31]–[33], we model the bursts of erasures (i.e. the memory of the channel) by a simple two-state Gilbert-Elliott channel (GEC) model. We then propose two algorithms that can enable a tradeoff for an improved balance between completion time and decoding delay of IDNC for different ranges of the channel memory.

With this introduction, we summarize the contributions and findings of our paper as follows: First, we present a holistic viewpoint of IDNC. We formulate the IDNC packet selection that enables a tradeoff for an improved balance between completion time and decoding delay for broadcast transmission over memoryless channels as an SSP problem. However, since finding such IDNC packet in the proposed SSP scheme is computationally complex, we use the SSP formulation and its geometric structure as guidelines for designing a new heuristic packet selection algorithm that efficiently improves the balance between completion time and decoding delay in IDNC systems. Second, we extend the proposed packet selection algorithm to erasure channels with memory and propose two different variations of the algorithm that take into account the channel memory conditions. Finally, we show that the proposed algorithm provides a more uniform decoding delay experience across all receivers.

The rest of this paper is organized as follows. The system

model is presented in Section II. The IDNC graph representation and packet generation are introduced in Section III. Section IV, presents the SSP problem formulation. In Section V, we present a geometric structure for the SSP problem that helps us with designing heuristic algorithms. A heuristic algorithm for IDNC packet selection is proposed in Section VI. The proposed heuristic algorithm is then extended to erasure channels with memory in Section VII, where also a new layered algorithm is introduced. Section VIII presents the simulation results. Finally, Section IX concludes the paper.

II. SYSTEM MODEL

The system model consists of a wireless sender that is required to deliver a block (denoted by $\mathcal{N}$) of N source packets to a set (denoted by $\mathcal{M}$) of M receivers. Each receiver is interested in receiving all the packets of $\mathcal{N}$. The sender initially transmits the N packets of the block uncoded in an *initial transmission phase*. Each sent packet is subject to erasure at receiver i with the probability p_i , $i \in \mathcal{M}$, which is assumed to be fixed during a block transmission period. Each receiver listens to all transmitted packets and feeds back a positive or negative acknowledgment (ACK or NAK) for each received or lost packet. At the end of the initial transmission phase, two “feedback sets” can be attributed to each receiver i :

- 1) The Has set (denoted by $\mathcal{H}_i$) is defined as the set of packets correctly received by receiver i .
- 2) The Wants set (denoted by $\mathcal{W}_i$) is defined as the set of packets that are missed at receiver i in the initial transmission phase of the current block. In other words, $\mathcal{W}_i = \mathcal{N} \setminus \mathcal{H}_i$.

The senders then stores this information in the *state feedback matrix* (SFM) $\mathbf{F} = [f_{ij}]$, $\forall i \in \mathcal{M}, j \in \mathcal{N}$ as:

$$f_{ij} = \begin{cases} 0 & j \in \mathcal{H}_i \\ 1 & j \in \mathcal{W}_i \end{cases} \quad (1)$$

Example 1: An example of SFM with $M = 4$ receivers and $N = 6$ packets is given as follows:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (2)$$

In this example, $f_{11} = 1$ denotes that packet 1 is missed at receiver 1, and $f_{21} = 0$ denotes that packet 1 is correctly received at receiver 2.

After the initial transmission phase, a *recovery transmission phase* starts, in which the sender exploits the diversity of received and lost packets to transmit network coded combinations of the source packets. Note that we denote the Wants and Has sets of receiver i at the start of the recovery transmission phase by $\mathcal{W}_i^s$ and $\mathcal{H}_i^s$, respectively. After each transmission, for each received/lost packet, the receivers send ACK/NAK to the sender. This information is then used by the sender to update the SFM. This process is repeated until all receivers obtain all packets. Similar two-phase transmission schemes have been widely considered in the literature for IDNC schemes [13]–[15], [24], [25], [28], [29]. We will

provide more comments about the two-phase transmission scheme in Section VIII.

Based on the Wants and Has sets information, in the recovery transmission phase, the transmitted coded packets can be one of the following options for each receiver i :

- 1) Non-innovative packet: A packet is non-innovative for receiver i if it contains no source packets from $\mathcal{W}_i$.
- 2) Instantly decodable packet: A packet is instantly decodable for receiver i if it contains only one source packet from $\mathcal{W}_i$. The set of receivers for which the transmitted packet is instantly decodable packet are referred to as the *targeted receivers*.
- 3) Non-instantly decodable packet: A packet is non-instantly decodable for receiver i if it contains two or more source packets from $\mathcal{W}_i$.

Example 2: For the SFM in (2), coded packet $1 \oplus 2$ is instantly decodable for all receivers as it consists of only one source packet from the Wants sets of all receivers. Thus, all receivers are targeted by this packet. However, packet $3 \oplus 4$ is only instantly decodable at receivers 1 and 4 (i.e. its targeted receivers are receivers 1 and 4). At receiver 2, packet $3 \oplus 4$ is non-instantly decodable, as it contains two source packets from receiver 2's Wants set. Furthermore, packet $3 \oplus 4$ is non-innovative at receiver 3 as it includes no source packet from receiver 3's Wants set.

We define the completion time and decoding delay similar to [5], [13]–[15], [25], [29] as follows:

Definition 1: Individual completion time (ICT) of receiver i , denoted by T_i^f , is the total number of transmissions required so that receiver i receives all its missing packets.

It should be noted that if receiver i is targeted by one of its missing packets in all transmissions, in the absence of packet erasures, T_i^f will be equal to the size of its Wants set at the start of the recovery transmission phase, i.e., $T_i^f = |\mathcal{W}_i^s|$.

Definition 2: Overall completion time (OCT), denoted by T^f , is the number of transmissions required so that all the receivers receive all their missing packets. In other words, the OCT is equal to the maximum ICT across all the receivers.

Definition 3: In time slot t , receiver i with non-empty Wants set experiences one unit of decoding delay, i.e. $d_i^t = 1$, if it successfully receives a packet that is either non-innovative or non-instantly decodable. If receiver i receives an instantly decodable packet it will not experience any decoding delay in this time-slot, i.e., $d_i^t = 0$.

Remark 1: Note that in this definition, we do not count channel inflicted delays due to erasures. The delay only counts “algorithmic” delays when we are not able to provide innovative and instantaneously decodable packets to a receiver.

Definition 4: In each time slot t , we define the accumulative decoding delay D_i^t to represent the summation of the decoding delays experienced by receiver i until time slot t . In other words, $D_i^t = \sum_{l=1}^t d_i^l$.

Two performance metrics that we take into account in this paper are the OCT and accumulative decoding delay. Our goal is to design efficient IDNC schemes that can improve the balance between these two performance metrics, as formulated in Section IV.

III. IDNC PACKET GENERATION

In this paper, we adopt IDNC [14], [15] as our NC transmission scheme. At targeted receivers, the IDNC packet is instantly decodable. However, at the rest of the receivers (referred to as *untargeted receivers*), the packet is either non-innovative or non-instantly decodable and is discarded, if successfully received. Thus, the untargeted receivers will experience one unit increase of their accumulative decoding delay.

We start this section by first exploring all possible packet combinations that are instantly decodable by any subset or if possible all receivers. All the feasible packet combinations can be represented as a graph, which was first used in the context of IDNC in [14], [15]. Then, we will briefly review the packet selection schemes in [15] and [14] that were used to separately improve IDNC's OCT and decoding delay, respectively.

As presented in [14], [15], the IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is constructed by first inducing a vertex $v_{ij} \in \mathcal{V}$ for each packet $j \in \mathcal{W}_i, \forall i \in \mathcal{M}$. In other words, any vertex v_{ij} represents a wanted packet j for receiver i . Two vertices v_{ij} and v_{kl} in $\mathcal{V}$ are connected by an edge $\mathcal{E}$ if any one of the following conditions is true:

C1: $j = l \Rightarrow$ The two vertices are induced by the loss of the same packet j by two different receivers i and k . An edge generated by this condition does not involve any combination, but expresses the interest of the two receivers in the same packet.

C2: $j \in \mathcal{H}_k$ and $l \in \mathcal{H}_i \Rightarrow$ The wanted packet corresponding to each vertex is in the Has set of the receiver of the other vertex. An edge generated by C2 represents a possible combination of packets j and l of the form $j \oplus l$ that will be instantly decodable for receivers i and k .

Given the graph formulation, the set of all feasible packet combinations in IDNC can be expressed as the set of packet combinations defined by all maximal cliques in $\mathcal{G}$ (a *maximal clique* is a clique that is not a subset of any larger clique). Consequently, the sender can generate an IDNC packet for a given transmission by XORing all the packets identified by the vertices of a selected maximal clique in $\mathcal{G}$. Assuming that κ is the selected maximal clique in $\mathcal{G}$, the targeted receivers of this clique are represented by $\mathcal{T}(\kappa)$.

The problem of improving the OCT of the IDNC scheme for broadcast erasure channels has been studied in [15] where it is shown that the expected ICT for receiver i , denoted by τ_i , if addressed in all future transmissions, can be expressed as $\tau_i = \frac{|\mathcal{W}_i|}{(1-p_i)}$. Having the expected ICT of all receivers calculated, it is shown in [15] that an efficient policy for reducing the OCT should select a maximal clique that includes a large number of vertices belonging to receivers having large τ_i values. In order to simplify such maximal clique selection in the IDNC graph $\mathcal{G}$, the authors in [15] proposed a maximum weight vertex search algorithm, where the weights of vertices in $\mathcal{G}$ signify the properties of their inducing receivers as follows. Let us define $a_{ij,kl}$ to be the adjacency indicator of vertices v_{ij} and v_{kl} in IDNC graph $\mathcal{G}$ such that:

$$a_{ij,kl} = \begin{cases} 1 & v_{ij} \text{ is connected to } v_{kl} \text{ in } \mathcal{G}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Given the adjacency indicator, the weighted degree Δ_{ij} of vertex v_{ij} in [15] is defined as $\Delta_{ij} = \sum_{\forall v_{kl} \in \mathcal{G}} a_{ij,kl} \tau_k$. Thus, the weight of vertex v_{ij} can be defined as $w_{ij} = \tau_i \Delta_{ij}$. This expression means that a vertex has a large weight when it both belongs to a receiver with large τ_i value and is connected to a large number of vertices having large τ_k values.

The problem of improving the decoding delay of IDNC scheme for broadcast erasure channels has been studied in [14] where it is shown that an efficient policy for reducing decoding delay is selecting a maximal clique that includes a large number of vertices belonging to receivers having high reception probabilities that are also connected to vertices with large reception probabilities (i.e. low erasure probabilities). Thus, the weight of vertex v_{ij} , w_{ij} , in [14] is defined as $w_{ij} = \Delta_{ij}(1-p_i)$, where Δ_{ij} signifies the connection of vertex v_{ij} to vertices having large reception probabilities and is defined in [14] as $\Delta_{ij} = \sum_{\forall v_{kl} \in \mathcal{G}} a_{ij,kl}(1-p_k)$.

Example 3: Let us again consider the SFM in (2). By using the technique in [15] and assuming no packet erasure occurs during the recovery transmission phase, the completion time is minimized if the packets are coded as: $1 \oplus 2$; 3 ; 6 ; 5 and 4 . Here, the packets are coded in such a way that the receiver(s) with the largest Wants set (i.e. receiver 2 in this example) is addressed by one of its missing packets in each transmission. However, this requirement may limit the number of receivers that can be targeted and as a result may increase the decoding delay. Under this scheme, the OCT of the block transmission is equal to 5 and the average decoding delay experienced by the receivers is equal to 1.25. However, if the scheme in [14] is adopted to reduce the decoding delay, the packets will be coded as: $1 \oplus 2$; $3 \oplus 4 \oplus 5$; 6 ; 3 ; 4 and 5 . In this scheme, in order to reduce the decoding delay, the maximum number of receivers should be targeted in each transmission. However, this may result in the receiver(s) with the largest Wants set to remain untargeted. Therefore, for this scheme, the OCT is equal to 6 and the average decoding delay experienced by the receivers is equal to 0.25. From this example, it can be easily seen that reducing the OCT on its own may result in an increased decoding delay and also reducing the decoding delay alone may result in an increased OCT of the transmission.

Unlike [15] and [14], in this study, our goal is to propose a new packet selection policy that can provide joint control of the OCT and decoding delay for IDNC schemes.

IV. PROBLEM FORMULATION

In this section, we present a holistic viewpoint of IDNC schemes in which the completion time and decoding delay are taken into account at the same time. By taking this viewpoint, we introduce a new IDNC scheme that enables a tradeoff for an improved balance between the OCT of the system and decoding delay performances of the receivers. It is worth noting that the notion of decoding delay considered in our proposed scheme is the accumulative decoding delay (defined in Definition 4). The key idea here is that the ICT of each receiver is not only determined by the number of its missing packets, but also the accumulative decoding delay that the respective receiver experiences. Furthermore, we note that the

OCT of the IDNC transmission is equal to the maximum of ICTs. The proposed scheme is formulated as an SSP problem.

Here, we first define $\mathbf{W}^s = [W_1^s, \dots, W_M^s]$ as the Wants vector, such that W_i^s is the cardinality of Wants set of receiver i , W_i^s , at the start of recovery phase. Furthermore, $\mathbf{D}^f = [D_1^f, \dots, D_M^f]$ is defined as the final accumulative decoding delay vector, where D_i^f is the final accumulative decoding delay experienced by receiver i (i.e. the accumulative decoding delay experienced by receiver i until it receives all its missing packets).

The best possible performance of IDNC in terms of the OCT and decoding delay can be achieved if in every single transmission all the receivers with non-empty Wants sets are targeted. In this case, after each transmission, assuming that no erasure occurs, the remaining number of transmissions is reduced by one and the accumulative decoding delays experienced by the receivers are zero. Under this scenario, the ICT of each receiver is equal to the size of its initial Wants set, W_i^s , and the OCT of the system is equal to the maximum ICT of the receivers (the size of the largest initial Wants set, i.e. $\max_{i \in \mathcal{M}} \{W_i^s\}$). Furthermore $D_i^f = 0, \forall i \in \mathcal{M}$.

However, since it may not always be possible to target all receivers with non-empty wants sets in every single IDNC transmission, due to instant decodability constraint, the receivers that are not targeted will experience a decoding delay, and thus, their ICTs will be increased by the value of their final accumulative decoding delay (i.e. the total number of the time-slots that they were not targeted). Therefore, we can write the ICT of receiver i , denoted by T_i^f , as

$$T_i^f = W_i^s + D_i^f, \quad i \in \mathcal{M} \quad (4)$$

As shown in (4), the ICT of each receiver depends on the size of its initial Wants set, W_i^s , and the final accumulative decoding delay it experiences, D_i^f . Having defined the receivers' ICTs, it can be easily inferred that OCT of the system is equal to the maximum ICT of the receivers, and can be expressed as

$$T^f = \max_{i \in \mathcal{M}} T_i^f = \max_{i \in \mathcal{M}} \{W_i^s + D_i^f\} \quad (5)$$

It is worth noting that based on (4), given W_i^s , minimizing the accumulative decoding delay of receiver i is equivalent to minimizing its ICT. Furthermore, based on (5), minimizing the OCT is equivalent to minimizing the largest ICTs. In the next subsection, we will show that the packet selection problem that offers a balance between the OCT and decoding delay of the receivers for the IDNC can be formulated as an SSP problem.

A. Stochastic Shortest Path (SSP) Problem

The SSP problem is a special case of an infinite horizon Markov decision process, which can model decision based stochastic dynamic systems with a terminating state. SSP problem was first used in the context of IDNC in [25] in order to select the packet combinations that can improve completion time. In SSP problem, different possible situations that the system could encounter are modeled as states $s \in \mathcal{S}$ (where $\mathcal{S}$ denotes the state space of the SSP problem). In each state $s \in \mathcal{S}$, the system must select an action a from an action space

$\mathcal{A}(s) \subseteq \mathcal{A}$ that will charge it an immediate cost $c(s, a, s')$ ($\mathcal{A}$ denotes the action space of the SSP problem). In the general form, the cost of a transition from state s to state s' is modeled as a scalar that depends on s , the taken action a , and s' . Under this scenario, in the SSP formulation, the expected cost $\bar{c}(s, a)$ is calculated as $\bar{c}(s, a) = \sum_{s' \in \mathcal{S}} P_a(s, s') c(s, a, s')$, where $P_a(s, s')$ represents the probability of system moving from state s to state s' once action a is taken. The terminating condition of the system can be thus represented as a zero-cost absorbing goal state. An SSP policy $\pi = [\pi(s)]$ is a mapping from $\mathcal{S} \rightarrow \mathcal{A}$ that associates a given action to each of the states. The optimal policy π^* of an SSP problem is the one that minimizes the cumulative mean cost until the goal state is reached.

The algorithms solving SSP problems define a value function $V_\pi(s)$ as the expected cumulative cost until absorption, when the system starts at state s and follows policy π . It can be recursively expressed for all $s \in \mathcal{S}$ as:

$$V_\pi(s) = \bar{c}(s, \pi(s)) + \sum_{s' \in \mathcal{S}(s, a)} P_\pi(s, s') V_\pi(s'), \quad (6)$$

where $\mathcal{S}(s, a)$ is the set of successor states to s when action a is taken (i.e. $\mathcal{S}(s, a) = \{s' | P_a(s, s') > 0\}$). Consequently, the optimal policy at state s can be defined for all $s \in \mathcal{S}$ as:

$$\pi^*(s) = \arg \min_{a \in \mathcal{A}(s)} \{ \bar{c}(s, a) + \sum_{s' \in \mathcal{S}(s, a)} P_a(s, s') V_{\pi^*}(s') \} \quad (7)$$

B. Problem Formulation using SSP Technique

In order to express the packet selection problem that improves the balance between the OCT and decoding delay of the system for IDNC in the form of an SSP problem, we define the following:

1) *State Space $\mathcal{S}$* : Each state s can be characterized by its Wants and accumulative decoding delay vectors, $\mathbf{W}(s) = [W_1(s), \dots, W_M(s)]$ and $\mathbf{D}(s) = [D_1(s), \dots, D_M(s)]$, respectively. The values of $\mathbf{W}(s)$ and $\mathbf{D}(s)$ at the starting state of the recovery transmission phase, s_s , are represented by $\mathbf{W}(s_s) = [W_1(s_s), \dots, W_M(s_s)]$ and $\mathbf{D}(s_s) = [0, \dots, 0]$, respectively, where $W_i(s_s) = W_i^s, \forall i \in \mathcal{M}$. Furthermore, we define the absorbing state, s_a , as the state in which all the receivers receive all their missing packets. In other words, the absorbing state is the final state of the recovery transmission phase in which $\mathbf{W}(s_a) = [0, \dots, 0]$. In addition, for each state s , we define $\mathcal{M}_w(s)$ to be the set of receivers who still need one or more packets. It is worth noting that the value of the accumulative decoding delay vector at the absorbing state; i.e. $\mathbf{D}(s_a) = [D_1(s_a), \dots, D_M(s_a)]$ where $D_i(s_a) = D_i^f, \forall i \in \mathcal{M}$; depends on the taken actions in all states prior to reaching the absorbing state.

2) *Action Spaces $\mathcal{A}(s)$* : For each state s , the action space $\mathcal{A}(s)$ consists of all possible maximal cliques in graph $\mathcal{G}(s)$ constructed from the SFM $\mathbf{F}(s)$ in state s . Defining $\mathcal{C}(s)$ as the set of maximal cliques in $\mathcal{G}(s)$, the cardinality of state s action space, i.e. $|\mathcal{A}(s)|$, is equal to $|\mathcal{C}(s)|$.

3) *State-Action Transitions Probabilities*: Here, we define the state value $U_i(s)$ for receiver i as

$$U_i(s) = W_i(s) + D_i(s) \quad (8)$$

Furthermore, the *state vector* for all receivers is defined as $\mathbf{U}(s) = [U_1(s), \dots, U_M(s)]$. Now, the state-action transition probability $P_a(s, s')$ for an action $a = \kappa(s) \in \mathcal{C}(s)$, can be defined based on the possibilities of the variations in $U_i(s)$ from state s to state s' .

To find $P_a(s, s')$, we first introduce the following three sets:

$$\mathcal{X} = \{i \in \mathcal{T}(\kappa(s)) \mid U_i(s') < U_i(s)\} \quad (9)$$

$$\mathcal{Y} = \{i \in \mathcal{M}_w(s) \setminus \mathcal{T}(\kappa(s)) \mid U_i(s') > U_i(s)\} \quad (10)$$

$$\mathcal{Z} = \{i \in \mathcal{M}_w(s) \mid U_i(s') = U_i(s)\} \quad (11)$$

where $\mathcal{M}_w(s)$ denotes all receivers with non-empty Wants sets at state s and $\mathcal{T}(\kappa(s))$ represents the set of all the targeted receivers in the maximal clique $\kappa(s)$. Here, the first set consists of the receivers who have been targeted by the clique $\kappa(s)$ and their $U_i(s)$ have been decreased from state s to state s' . This means that these receivers have successfully received an IDNC packet, which has addressed them by one of their missing packets. Thus, the size of their Wants sets is reduced and their accumulative decoding delays are remained unchanged. The second set includes the receivers who have not been targeted but have successfully received the transmitted packet. In this case $U_i(s)$ is increased from state s to state s' , since the Wants sets of these receivers have remained unchanged and their accumulative decoding delays have increased due to successfully receiving either a non-innovative or a non-instantly decodable packet. The third set includes the receivers who have not received any packet due to packet erasure and as a result, their Wants sets and accumulative decoding delays have remained unchanged, thus $U_i(s') = U_i(s)$. Based on the definitions of these three sets, $P_a(s, s')$ can be expressed as follows:

$$P_a(s, s') = \prod_{i \in \mathcal{X}} (1 - p_i) \cdot \prod_{i \in \mathcal{Y}} (1 - p_i) \cdot \prod_{i \in \mathcal{Z}} p_i \quad (12)$$

Example 4: Let us consider the following SFM with $M = 2$ receivers and $N = 4$ packets:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad (13)$$

The state representation and action space for this SFM are depicted in Figure 1. In this figure, the actions are represented by a_i , and action $a_i = j$ refers to the transmission of IDNC packet j . Furthermore, Figure 1 also shows the state-action transitions probabilities and their corresponding resulting states given that action a_1 is performed.

4) *State-Action Costs:* The best possible action is the action that addresses all the receivers with non-empty Wants sets at state s , denoted by $\mathcal{M}_w(s)$, by one of their missing packets. Under this scenario, assuming no erasure occurs, the Wants sets of all the receivers are reduced from state s to state s' and their accumulative decoding delays remain unchanged (i.e. $W_i(s') = W_i(s) - 1$ and $D_i(s') = D_i(s), \forall i \in \mathcal{M}_w(s)$). In this case, for each receiver i we will have $U_i(s') - U_i(s) = -1, \forall i \in \mathcal{M}_w(s)$. This is the best performance that can be achieved for an IDNC scheme.

Knowing that any transition (due to any action) takes one packet transmission, the cost of action a on each receiver i can be defined as $c_i(s, a, s') = 1 + (U_i(s') - U_i(s))$. This

Fig. 1: State representation, action space and the possible transitions for action a_1 of the example SFM in (13).

results in three possible cost values, i.e. $\{0, 1, 2\}$, associated with action a on receiver i that can be expressed as follows:

- $C_i(s, a, s') = 0$ means that action a does not incur any cost on receiver i in terms of its Wants set and accumulative decoding delay, if it successfully receives one of its missing packets. In this case, $U_i(s') - U_i(s) = -1$ and $c_i(s, a, s') = 1 + (U_i(s') - U_i(s)) = 1 + (-1) = 0$.
- $c_i(s, a, s') = 1$ means that receiver i (targeted/untargeted) did not receive the coded packet due to packet erasure. In this case, there is no cost on the accumulative decoding delay, however, the Wants set of receiver i remains unchanged, as no missing packet was decoded. Here, at least one more time-slot (one transmission) is required to be able to reduce the size of receiver i 's Wants set. Under this scenario, $U_i(s') - U_i(s) = 0$ and $c_i(s, a, s') = 1 + (U_i(s') - U_i(s)) = 1 + 0 = 1$.
- $c_i(s, a, s') = 2$ means that receiver i was not targeted by action a and has successfully received either a non-instantly decodable or a non-innovative packet. In this case, there are costs on both the accumulative decoding delay and Wants set of receiver i , as it experiences an increase in its accumulative decoding delay and the size of its Wants set remains unchanged. As a result $U_i(s') - U_i(s) = 1$ and $c_i(s, a, s') = 1 + (U_i(s') - U_i(s)) = 1 + 1 = 2$.

Based on the above discussion, if receiver i is targeted by action a , i.e. $i \in \mathcal{T}(a)$, the cost will be

$$c_i(s, a, s')|_{i \in \mathcal{T}(a)} = \begin{cases} 0 & \text{with prob. } (1 - p_i) \\ 1 & \text{with prob. } p_i \end{cases} \quad (14)$$

Thus, the expected cost given receiver i is targeted by action a can be calculated as

$$\bar{c}_i(s, a | i \in \mathcal{T}(a)) = 0 \times (1 - p_i) + 1 \times p_i = p_i \quad (15)$$

However, if receiver i is not targeted by action a , i.e. $i \notin \mathcal{T}(a)$, the cost will be

$$c_i(s, a, s')|_{i \notin \mathcal{T}(a)} = \begin{cases} 1 & \text{with prob. } p_i \\ 2 & \text{with prob. } (1 - p_i) \end{cases} \quad (16)$$

Thus, the expected cost given receiver i is not targeted by action a can be calculated as

$$\bar{c}_i(s, a | i \notin \mathcal{T}(a)) = 1 \times p_i + 2 \times (1 - p_i) = 2 - p_i \quad (17)$$

Fig. 2: Geometric structure of SFM in (13).

The total expected cost of action a over all the receivers in $\mathcal{M}_w(s)$ can thus be defined as

$$\begin{aligned} \bar{c}(s, a) &= \sum_{i \in \mathcal{M}_w(s)} \bar{c}_i(s, a |_{i \in \mathcal{T}(a)}) + \sum_{i \in \mathcal{M}_w(s)} \bar{c}_i(s, a |_{i \notin \mathcal{T}(a)}) \\ &= \sum_{i \in \mathcal{T}(a)} p_i + \sum_{i \in \{\mathcal{M}_w(s) \setminus \mathcal{T}(a)\}} (2 - p_i) \end{aligned} \quad (18)$$

5) *Optimal Policy*: The optimal policy as presented in Section IV-A can be expressed as

$$\begin{aligned} \pi^*(s) &= \arg \min_{a \in \mathcal{A}(s)} \{ \bar{c}(s, a) + \sum_{s' \in \mathcal{S}(s, a)} P_a(s, s') V_{\pi^*}(s') \} \\ &= \arg \min_{a \in \mathcal{A}(s)} \{ \bar{c}(s, a) + \mathbb{E}_a[V_{\pi^*}(s')] \} \end{aligned} \quad (19)$$

where $\mathbb{E}_a$ is the expectation operator over different transmission probabilities when action a is taken. Thus, the optimal action at state s is the action that minimizes the cost as well as the expectation of the optimal value functions of the successor states. However, solving this SSP problem is computationally complex and requires exhaustive iterative techniques [34]. Furthermore, there is no closed-form solution to this problem. Thus, we will study the geometric structure of the SSP solution in (19). In other words, our aim of the SSP formulation is not to use it as a solution, but to study its properties by the help of its geometric structure and find some guidelines for policies that can improve the balance between the OCT and decoding delay in IDNC systems. We then use these policies to design simple yet efficient heuristic algorithms in Section VI.

V. GEOMETRIC STRUCTURE OF THE PROBLEM

In order to find some guidelines for the policies that can efficiently improve the balance between the OCT and decoding delay in IDNC systems, in this section, we study the geometric structure of the SSP problem. Given the representation of the SSP problem in each state s by the state vector of the receivers $\mathbf{U}(s) = [U_1(s), \dots, U_M(s)]$, we can now explain the geometric structure of the problem as follows. First, we consider an M -dimensional Cartesian space, and assign to each point $\Delta = [\delta_1, \dots, \delta_M]$ in this space all the states that have the state vectors $\mathbf{U}(s) = [U_1(s), \dots, U_M(s)]$ equal to the coordination of this point. Although many states can have the same state vector, these states differ from one another by their SFMs. The absorbing state is the state for which all $W_i(s) = 0, \forall i \in \mathcal{M}$.

Under the special scenario where the accumulative decoding delays are zero, i.e. $D_i(s) = 0, \forall i \in \mathcal{M}$, the absorbing state is located in the origin of the considered M -dimensional Cartesian space. However, in general, the decoding delays experienced by the receivers until arriving at the absorbing state can be non-zero positive integers, and consequently the absorbing point will not necessarily be located in the origin of the space.

After each transmission, if the packet is successfully received at a receiver, there are two possibilities, 1) it is instantly decodable, and thus $U_i(s') = U_i(s) - 1$, 2) it is either non-instantly decodable or non-innovative, and thus $U_i(s') = U_i(s) + 1$. However, if the packet is not received at the receiver, then $U_i(s') = U_i(s)$. Therefore, it can be easily concluded that the system can at most move from point $\Delta = \mathbf{U}(s)$ to another point $\Delta' = \mathbf{U}(s')$ which is a vertex in the hypercube $\Gamma(s)$ defined as:

$$\Gamma(s) = \{ \Delta' | U_i(s') - U_i(s) \in \{-1, 0, 1\}, \forall i \in \mathcal{M}_w \} \quad (20)$$

In other words, $\Gamma(s)$ is the hypercube of side length 1, in which $\mathbf{U}(s)$ and $\mathbf{U}(s')$ are two of the corners.

Here, we start with the geometric structure of the SSP problem in the erasure-free case and then extend it to the case with erasures.

Case1: Erasure-free Case

In the erasure-free case, since transmitted packets are always successfully received by the receivers, depending on the received packet being instantly decodable or not, we will have $U_i(s') = U_i(s) - 1$ or $U_i(s') = U_i(s) + 1$. Under this scenario, $\mathbf{U}(s)$ and $\mathbf{U}(s')$ are always two diagonal corners in the hypercube $\Gamma(s)$, i.e. $\Gamma(s) = \{ \Delta' | U_i(s') - U_i(s) \in \{-1, 1\}, \forall i \in \mathcal{M}_w \}$.

Subcase 1. There exists an action with zero total cost:

Under the erasure-free scenario, it can be shown that at any state s choosing the action that transitions the system to the opposite diagonal point in the M_w -dimensional hypercube, for which $U_i(s') - U_i(s) = -1$ and thus $c_i(s, a) = 0, \forall i \in \mathcal{M}_w$, would not adversely affect the optimality of future decisions. This is due to the fact that all packets in this action will be received by all receivers and therefore, they would not contribute to any future cost.

Example 5: Figure 2 illustrates the geometric structure of SFM in (13). In this example, there exist three actions, actions a_4, a_5 and a_6 , that target both receivers (i.e. for these actions $U_1(s') - U_1(s) = U_2(s') - U_2(s) = -1$), and thus their total costs are zero. Furthermore, these actions give the chance to the system to reach absorption with two more transmissions, which makes them optimal actions.

However, such zero-cost actions may not always exist, due to the instant decodability constraint. Consequently, we need a method to find efficient actions that provide an improved balance between the OCT and decoding delay.

Subcase 2. There does not exist an action with zero total cost: In the absence of an action with zero total cost, in order to find efficient actions that provide an improved balance between the OCT and decoding delay, we consider the geometric structure in the following example.

Example 6: Referring to the geometric structure of SFM in (13), as illustrated in Figure 2, let us assume that the only available actions are actions a_1, a_2 and a_3 . All these actions only target one receiver and thus, the untargeted receiver will experience a unit increase in its accumulative decoding delay. For these actions we have $c(a_1, s) = c(a_2, s) = c(a_3, s) = 2$. Although these actions have equal costs and perform equally in terms of the decoding delay, but actions a_1 and a_3 are preferred over action a_2 in terms of OCT, as they target the receiver with the largest Wants set (i.e. receiver 1) and thus, bring the IDNC one step closer to block completion. The superiority of actions a_1 and a_3 over action a_2 and their closeness to the absorption is shown through smaller geometric distance of point $[2, 3]$ from the origin (point $[0, 0]$), compared to point $[4, 1]$.

It is worth noting that in the above example, the L_2 norm (Euclidian distance) is used to represent a state's closeness to the origin.

The above discussion can be summarized as the following remark.

Remark 2: [Design Guideline] Based on the studied geometric structure of the SSP problem, at any state s , the geometric distances of the actions' resulting points from the origin reflect the efficiency of those actions. In other words, the actions that bring the system closest to the origin result in reaching the completion faster with lower decoding delays. Furthermore, we can conclude that targeting the receiver with the maximum state value, i.e. minimizing the maximum entry of the state vector, brings the system closest to the origin faster. This is also reflected in the geometric distance of the destination points from the origin.

Furthermore, it can also be easily inferred that giving higher priorities to receivers with larger values of $U_i(s) = W_i(s) + D_i(s)$ can potentially result in a lower variance of the decoding delay experienced by the receivers in the system. It means that when the decoding delay of a receiver increases, the value of $U_i(s)$ also increases, and as a result of that the respective receiver will be given a higher priority. This can improve the decoding delay fairness among the receivers. The simulation results on the variance of the decoding delay across all receivers are represented in Section VIII.

Case 2: Erasure Case

Due to the nature of wireless broadcast systems and the fact that the SFM changes probabilistically after each transmission as a result of packet erasures, in this paper, we design the IDNC packet dynamically according to the received feedback in each time slot. Under this scenario, since the packet erasures are not known ahead of time, our approach is a greedy-based algorithm in which at each transmission based on the updated SFM, a single coded packet is designed (guided by Remark 2 above).

For erasure channels, the effect of packet erasures should be reflected on the geometric structure of the problem. Let i and k be two receivers having the same Wants set size, but $p_i > p_k$. Consequently, receiver i will require on average more targeting attempts compared to receiver k in order to deplete its Wants set. Since we assume that erasure probabilities do not

change during the transmission of a block, targeting receiver k and ignoring receiver i is expected to result in a higher OCT, especially when $U_i(s)$ is among the largest values in $\mathbf{U}(s)$. According to these observations, receiver i should be given a higher priority of service than receiver k .

In order to implement the above prioritization, we define a channel weighted Wants value as $\tilde{W}_i(s) = \frac{W_i(s)}{1-p_i}$, and consequently $\tilde{\mathbf{U}}(s) = [\tilde{U}_1(s), \dots, \tilde{U}_M(s)]$, where

$$\tilde{U}_i(s) = \tilde{W}_i(s) + D_i(s) = \frac{W_i(s)}{1-p_i} + D_i(s) \quad (21)$$

Based on this new vector definition, we can re-define our space such that the points Δ are identified by the coordinates of the vectors $\tilde{\mathbf{U}}(s)$ instead of $\mathbf{U}(s)$, $\forall s \in \mathcal{S}$. In this case, the actions move the system within hyper-rectangles $\Gamma'(s)$ with sides either equal to 1 or $\frac{1}{1-p_i}$ in the i -th dimension. It means if an action results in an increase in the accumulative decoding delay, then $\tilde{U}_i(s') - \tilde{U}_i(s) = 1$, however, if it addresses one of the receiver i 's missing packets, it leads to $\tilde{U}_i(s') - \tilde{U}_i(s) = -\frac{1}{1-p_i}$. Moreover, if receiver i does not receive the packet due to erasure, then $\tilde{U}_i(s') - \tilde{U}_i(s) = 0$. In other words:

$$\Gamma'(s) = \{\Delta' | \tilde{U}_i(s') - \tilde{U}_i(s) \in \{-\frac{1}{1-p_i}, 0, 1\}, \forall i \in \mathcal{M}\} \quad (22)$$

In the next section, with the help of the above-mentioned design guidelines, we will propose a heuristic packet selection algorithm.

VI. HEURISTIC ALGORITHM FOR PACKET SELECTION

In this section, we propose a greedy algorithm to select a maximal clique in the IDNC graph according to the findings in the previous section. We use L_2 norm here, but other norms are also possible. The proposed algorithm performs clique selection using a maximum weight vertex search approach. For this search to be efficient in finding maximal cliques, the vertices' weights must not only reflect the $(\tilde{U}_i(s))^2$ values of their inducing receivers, but also their adjacency to the vertices with high $(\tilde{U}_k(s))^2$.

We define the weighted degree of vertex v_{ij} , denoted by $\Theta_{ij}(s)$, as:

$$\Theta_{ij}(s) = \sum_{\forall v_{kl} \in \mathcal{G}(s)} a_{ij,kl} (\tilde{U}_k(s))^2 \quad (23)$$

where $a_{ij,kl}$ was defined in (3). Thus, a large weighted degree reflects its adjacency to a large number of vertices belonging to receivers with large values of $(\tilde{U}_k(s))^2$. We finally design the vertex weight $w_{ij}(s)$ for vertex v_{ij} as:

$$w_{ij}(s) = (\tilde{U}_i(s))^2 \Theta_{ij}(s) \quad (24)$$

Consequently, a vertex has a high weight if it both belongs to a receiver with large $(\tilde{U}_i(s))^2$, and is connected to the receivers with large $(\tilde{U}_k(s))^2$ values.

Based on the above weight definition, we introduce our proposed packet selection algorithm as follows. In each state s , the algorithm starts by selecting the vertex with the maximum weight, denoted by v^* , and adds it to the clique κ^* . Note that at first, κ^* is an empty set. Then at each following iteration, the algorithm first recomputes the new vertices' weights within the subgraph connected to all previously selected vertices

in κ^* , denoted by $\mathcal{G}_{\kappa^*}(s)$, then adds the new vertex with the maximum weight to it. The algorithm stops when there is no further vertex connected to all vertices in κ^* . We refer to this algorithm as *maximum weight vertex search algorithm* (MWVS). The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed MWVS Algorithm

- 1) **Initialize** $\kappa^*(s) = \emptyset$
Construct $\mathcal{G}(s)$ based on $\mathbf{F}(s)$.
 - 2) **While** $\mathcal{G}(\kappa^*(s)) \neq \emptyset$ **do**
Compute $w_{ij}(s), \forall v_{ij} \in \mathcal{G}(\kappa^*(s))$ using (3), (23) and (24).
Select $v^* = \arg \max_{v_{kl} \in \mathcal{G}(\kappa^*(s))} \{w_{kl}(s)\}$.
Set $\kappa^*(s) \leftarrow \kappa^*(s) \cup v^*$.
Update subgraph $\mathcal{G}(\kappa^*(s))$.
-

It is worth noting that in the proposed MWVS algorithm, one can consider a more general state value $\tilde{U}_i(s)$ in (23), where $\tilde{U}_i(s)$ is expressed as

$$U_i(s) = \lambda \tilde{W}_i(s) + (1 - \lambda) D_i(s) = \frac{\lambda W_i(s)}{1 - p_i} + (1 - \lambda) D_i(s) \quad (25)$$

That is, different weights are considered for the Wants set and the accumulative decoding delay. Under this scenario, if $\lambda = 1$, the above algorithm will turn into the algorithm proposed in [15] that aims at reducing the completion time in each transmission, regardless of receivers' decoding delays. If $\lambda = 0$, the above algorithm will turn into an algorithm that only aims at reducing the accumulative decoding delays of the receivers in each transmission while trying to improve the decoding delay fairness across them. For any $0 < \lambda < 1$, the proposed algorithm will enable a tradeoff between these two extreme cases. In the rest of this paper, we assign equal weights to $\frac{W_i(s)}{1 - p_i}$ and $D_i(s)$ in the proposed MWVS algorithm, unless stated otherwise.

VII. HEURISTIC PACKET SELECTION ALGORITHM FOR ERASURE CHANNELS WITH MEMORY

In this section, our goal is to extend our proposed MWVS scheme to IDNC transmissions in erasure channels with memory. To model erasure channels with memory, we employ the well-known Gilbert-Elliott channel (GEC) [31] which is a Markov model with a *good* and a *bad* state. When the channel is in the good state packets can be successfully received, and when the channel is in the bad state packets are lost (e.g., due to deep fades in the channel). The probability of moving from the good state G to the bad state B is $b \triangleq \Pr(G \rightarrow B)$ and the probability of moving from the bad state B to the good state G is $g \triangleq \Pr(B \rightarrow G)$. Steady-state probabilities are derived as $P_G \triangleq \Pr(C_i = G) = \frac{g}{b+g}$ and $P_B \triangleq \Pr(C_i = B) = \frac{b}{b+g}$, where C_i is the channel state of receiver i in the previous transmission. Here, without loss of generality, we assume that $0 < b = g \leq 0.5$, which results in equiprobable states in the steady-state regime. Other scenarios can be considered in a similar manner. Following [31], we define the memory content of the GEC as $0 \leq \mu = 1 - b - g < 1$, which signifies the persistence of the channel in remaining in the same state. A

small μ means a channel with little memory and a large μ means a channel with large memory. We assume that different receivers' links are independent of each other with the same state transition probabilities.

A. Maximum Weight Vertex Search Algorithm (MWVS) for Channels with Memory

Here, the proposed MWVS algorithm in Section VI is modified so that it takes into account the channel memory conditions. In the modified framework, the positive or negative acknowledgment (ACK or NAK) that each receiver feeds back for each received or lost packet can be utilized to infer the channel state of that receiver in the previous transmission. The proposed MWVS algorithm in Section VI can then be generalized for erasure channels with memory by defining the probability of successful reception by receiver i as the probability of moving to the good state G in the current time-slot from its previous state C_i , i.e. $\Pr(C_i \rightarrow G)$. So the proposed MWVS algorithm can be easily implemented in erasure channels with memory by replacing $1 - p_i$ with $\Pr(C_i \rightarrow G)$ in (21) as

$$\tilde{U}_i(s) = \frac{W_i(s)}{\Pr(C_i \rightarrow G)} + D_i(s) \quad (26)$$

In other words, the weight of each vertex in (24) can now be recalculated based on the conditional reception probability of its inducing receiver, given its previous state, as

$$\begin{aligned} w_{ij}(s) &= (\tilde{U}_i(s))^2 \Theta_{i,j}(s) \\ &= \left[\frac{W_i(s)}{\Pr(C_i \rightarrow G)} + D_i(s) \right]^2 \Theta_{i,j}(s) \end{aligned} \quad (27)$$

However, for erasure channels with strong memory, the receivers have a strong tendency to stay in their previous states. It means if they have been in state G in the previous time-slot, they are most likely to stay in state G in the current time-slot, and vice versa, if they have been in state B , they are most likely to stay in state B . Under this case, for the receivers in state B , $\Pr(B \rightarrow G)$ will be very small and as a result the term $\frac{W_i(s)}{\Pr(C_i \rightarrow G)}$ in (27) will be large. Consequently, high weights will be given to the receivers that have been in state B in the previous transmission (also referred to as *bad-channel receivers* (BCR)). But it should be noted that targeting the BCRs most likely would not result in any decoding for them, as with a very high probability their channels will remain in state B in the current transmission. However, addressing the receivers that were in state G in the previous transmission (also referred to as *good-channel receivers* (GCR)) can potentially result in the decoding of their missing packets. Inspired by these scenarios, in the next sub-section, we will introduce a *layered maximum weight vertex search algorithm* (referred to as MWVS-Layered), which is specifically designed for erasure channels with persistent memory.

B. Layered Maximum Weight Vertex Search Algorithm (MWVS-Layered)

Here, our goal is to extend the proposed MWVS algorithm in Section VII-A for erasure channels with persistent memory. In order to do so, we follow the same approach as in [33].

The proposed algorithm comprises two different layers of subgraphs. The first layer of subgraph, $\mathcal{G}_g(s) \subseteq \mathcal{G}(s)$, consists of vertices of GCRs. In the first step, the MWVS algorithm is applied on the subgraph $\mathcal{G}_g(s)$, and $\kappa_g^*(s)$ is obtained. Then, in the second step, the algorithm finds $\kappa_b^*(s)$ by applying the MWVS algorithm another time on the second layer of subgraph, $\mathcal{G}_b(s)$, consisting of BCRs that are adjacent to all the vertices of the chosen clique $\kappa_g^*(s)$. Thus, the final clique can be obtained by the union of the cliques from the two layers as $\kappa^*(s) = \kappa_g^*(s) \cup \kappa_b^*(s)$. The steps of MWVS-Layered algorithm are summarized in Algorithm 2.

Algorithm 2 Proposed MWVS-Layered Algorithm

- 1) **Initialize** $\kappa_g^*(s) = \emptyset$ and $\kappa_b^*(s) = \emptyset$
Construct $\mathcal{G}(s)$ based on $\mathbf{F}(s)$.
Form $\mathcal{G}_g(s)$ and $\mathcal{G}_b(s)$ according to the channels' previous states $C_i, \forall i \in \mathcal{M}$.
 - 2) **While** $\mathcal{G}_g(\kappa_g^*(s)) \neq \emptyset$ do
Compute $w_{ij}(s), \forall v_{ij} \in \mathcal{G}_g(\kappa_g^*(s))$ using (3), (23) and (27).
Select $v^* = \arg \max_{v_{kl} \in \mathcal{G}_g(\kappa_g^*(s))} \{w_{kl}(s)\}$.
Set $\kappa_g^*(s) \leftarrow \kappa_g^*(s) \cup v^*$.
Update subgraphs $\mathcal{G}_g(\kappa_g^*(s))$ and $\mathcal{G}_b(\kappa_b^*(s))$.
 - 3) **While** $\mathcal{G}_b(\kappa_b^*(s)) \neq \emptyset$ do
Compute $w_{ij}(s), \forall v_{ij} \in \mathcal{G}_b(\kappa_b^*(s))$ using (27).
Select $v^* = \arg \max_{v_{kl} \in \mathcal{G}_b(\kappa_b^*(s))} \{w_{kl}(s)\}$.
Set $\kappa_b^*(s) \leftarrow \kappa_b^*(s) \cup v^*$.
Update subgraph $\mathcal{G}_b(\kappa_b^*(s))$.
 - 4) $\kappa^*(s) = \kappa_g^*(s) \cup \kappa_b^*(s)$
-

VIII. SIMULATION RESULTS

In this section, we present the simulation results comparing the performance of our proposed MWVS and MWVS-Layered algorithms and the schemes in [14], [15], [33] over a wide range of channel memory conditions. As our benchmark for the minimum OCT performance, we will compare the OCT of our proposed MWVS and MWVS-Layered algorithms with the RLNC scheme proposed in [6].

Furthermore, we provide comparison results with the schemes in [23], [24]. However, as these schemes are not directly applicable to our system model, where we consider a general number of receivers and a finite-length block of packets, we have modified these scheme as follows:

- In order to be able to compare with [24], as no explicit algorithm or technique for choosing the coded packet that addresses the maximum number of receivers in general case is proposed, we have modeled addressing the maximum number of receivers with an instantly decodable packet in the recovery transmission phase as finding the maximum clique¹ in our IDNC graph and transmitting the coded combination corresponding to this maximum clique.
- In order to be able to compare our proposed scheme with [23] and [26], we first need to modify it in such a way

¹Maximum clique in a graph is a clique with the maximum number of fully connected vertices. In other words, maximum clique is the largest maximal clique in the graph.

that it can be employed in the broadcast of a finite-length block of packets to a general number of receivers. To this end, we perform the following steps:

- 1) If there exists a coded combination that targets all receivers, send that combination, otherwise send a new uncoded packet.
- 2) However, if there does not exist a new uncoded packet to be transmitted, send the packet combination that addresses the maximum number of receivers (equivalent to the maximum clique in our GIDNC graph).

From the above steps, it can be easily inferred that this scheme considers a single-phase transmission, where mixed new uncoded and coded packets are transmitted.²

As will be shown in the simulation results, addressing the maximum number of receivers in these modified versions of the schemes in [23], [24], [26], does not necessarily reduce the OCT and/or decoding delay of the system efficiently. This is due to the fact that in order to efficiently reduce the OCT, a maximal clique including a large number of receivers with large Wants sets and high erasure probabilities should be selected for transmission [15]. Also in order to efficiently reduce the decoding delay, a maximal clique that includes a large number of receivers with high probabilities of successful reception (low erasure probabilities) should be chosen for transmission [14]. However, selecting the maximum clique does not necessarily guarantee these desired properties.

We start with our simulation results for memoryless erasure channels and compare the performance of our proposed MWVS algorithm with the schemes in [15], [14], [24], and [23], denoted by “Min-OCT”, “Min-DD”, “Max-Clique”, and “Mixed”, respectively. Furthermore, simulation results for $\lambda = 0$ and 1 in (25), denoted by “MWVS ($\lambda = 0$)” and “MWVS ($\lambda = 1$)”, respectively, are provided.

In our simulations for the broadcast memoryless erasure channels, we assume that packet erasures of different receivers change from block to block in the range $[0.05, 0.3]$ with an average equal to 0.15. The simulations are performed for different number of packets and receivers in the system. It should be noted that the presented simulation results in this section are the mean values, i.e. the OCT results show the average OCT of the transmission of N packets over 500 instances of SFM. In terms of the decoding delay, the mean decoding delay of different receivers are computed per block, and then these mean decoding delays are averaged over 500 instances of SFM. Hence, the decoding delay results are actually the mean of mean decoding delays.

Figure 3(a) depicts the OCT and decoding delay tradeoff curves of different algorithms for various number of packets N for $M = 30$ receivers. Moreover, the OCT and decoding delay tradeoff curves of these algorithms for various number of receivers M for $N = 30$ packets are presented in Figure 3(b).

²The proposed scheme in this paper can also be adapted to a single-phase transmission framework by starting from an all-ones SFM and setting $W_i(s_s) = N, \forall i \in \mathcal{N}$. However, the single-phase transmission scheme is not expected to further improve the OCT and decoding delay performance of the two-phase transmission scheme, as the initial phase of transmissions in the latter, i.e., uncoded transmission of the packets in the block once, is already optimal both in terms of the OCT and decoding delay at the same time.

(a)

(b)

Fig. 3: OCT versus Decoding delay (a) for different number of packets N and $M = 30$ receivers, and (b) for different number of receivers M and $N = 30$ packets.

From these figures, we first observe that the Min-OCT algorithm in [15] that achieves the minimum OCT among the IDNC schemes in Figures 3(a) and 3(b), results in a rather high mean accumulative decoding delay, and the Min-DD algorithm in [14] that achieves the minimum decoding delay performance, results in a rather high OCT. However, in these figures it is shown that our proposed MWVS algorithm provides an improved balance between the OCT and decoding delay for the whole range of number of packets and receivers. Furthermore, as it was expected, we observe that the performance of the proposed MWVS algorithm with $\lambda = 1$ is the same as the performance of Min-OCT algorithm proposed in [15]. Also, it can be seen that the performance of the proposed algorithm with $\lambda = 0$ is very close to the performance of the Min-DD algorithm proposed in [14]. However, it is worth noting that the proposed MWVS algorithm when $\lambda = 0$ aims to reduce the accumulative decoding delay (defined in Definition 4) while improving the decoding delay fairness across the receivers.

Fig. 4: Variance of the decoding delay versus number of packets N for $M = 30$ receivers.

However, the Min-DD algorithm in [14] aims to reduce the decoding delay of the system in each time-slot (defined in Definition 3) without taking into account receivers' decoding delay fairness.

Figure 4 illustrates the variance of the decoding delay, as a measure of decoding delay fairness, versus the number of packets N for $M = 30$ receivers. From this figure, it can be seen that our proposed MWVS algorithm significantly outperforms the other algorithms in terms of decoding delay fairness.

For erasure channels with memory, the full graph search and the layered graph search algorithms proposed in [33] are used as our reference for the minimum decoding delay performance. These algorithms are denoted by "Min-DD" and "Min-DD-Layered" in the figures, respectively.

As our reference for the minimum OCT performance for erasure channels with memory, we have modified the algorithm in [15] to become channel memory aware by replacing the probability of successful reception at receiver i with $Pr(C_i \rightarrow G)$. We refer to this scheme as "Min-OCT". Furthermore, we have extended this scheme to a two-layered algorithm, where the first layer consists of GCRs and the second layer consists of BCRs. In the first step, the algorithm is applied on the first layer and a clique of GCRs is obtained. Then, in the second step, the algorithm is applied to the second layer and a clique of BCRs that are adjacent to all the vertices of the chosen clique of GCRs is found. Then, the final clique is obtained by the union of the cliques from the two layers. In our simulation results, this scheme is referred to as "Min-OCT-Layered".

For the broadcast erasure channels with memory, we assume $b_i = g_i = b$ for all the receivers, and the channel memory, $\mu = 1 - b - g = 1 - 2b$, ranges from 0 (memoryless) to 0.98 (very persistent memory). The simulation results are provided for a wide range of channel memory contents as well as different number of packets and receivers.

Figures 5(a) and 5(b) illustrate the OCT of the receivers versus channel memory for $N = M = 30$ packets and receivers, respectively. As can be seen from these figures,

(a)

(b)

Fig. 5: OCT versus channel memory for $N = M = 30$ packets and receivers, (a) $0 \leq \mu \leq 0.8$, (b) $0.8 \leq \mu \leq 0.98$.

for low channel memory content (roughly ranging from 0 to 0.45), the full graph algorithms outperform their layered graph counterparts in terms of OCT. However, when the memory content of the channel is high (roughly ranging from 0.45-0.98), the layered graph techniques significantly outperform their full graph counterparts. The mean decoding delay performance versus channel memory is depicted in Figure 6. From this figure we can see that in terms of the decoding delay, the Min-DD algorithm outperforms the Min-DD-Layered for memory content ranging from 0 to 0.5, while the Min-DD-Layered outperforms Min-DD for higher channel memory contents (ranging from 0.5 to 0.98). For all the other investigated schemes, the layered graph techniques always result in lower decoding delays compared to their full graph counterparts. This is due to the fact that in the layered graph techniques, the priority is always given to the GCRs to be addressed by one of their missing packets, and as shown in [14] giving higher priorities to the receivers with higher probabilities of successful reception improves the decoding

Fig. 6: Decoding delay versus channel memory for $N = M = 30$ packets and receivers.

delay experienced by the receivers. Furthermore, as shown in these figures, the proposed MWVS-Layered scheme provides a better balance between the OCT and decoding delay for the whole range of channel memory content.

Figures 7(a) shows the OCT and decoding delay tradeoff curves of the system for different number of packets N for channel memory $\mu = 0.6$ and $M = 30$. The results show that for $\mu = 0.6$ the layered graph techniques outperform their full graph counterparts. Again it can be seen that the Min-OCT-Layered algorithm that achieves the lowest OCT among the IDNC schemes results in the worst mean decoding delay among the layered graph algorithms, and the Min-DD-Layered algorithm that achieves the lowest decoding delay results in the worst OCT. However, for $\mu = 0.6$ as we expected, the proposed MWVS-Layered algorithm results in an improved balance between the OCT and decoding delay. Figure 7(b) depicts the variance of the decoding delay versus number of packets N for channel memory $\mu = 0.6$ and $M = 30$ receivers. As depicted, the proposed MWVS-Layered algorithm always outperforms the other algorithms in terms of the variance of the decoding delay. This can be translated into a better fairness in the decoding delay experienced by different receivers when MWVS-Layered algorithm is used.

IX. CONCLUSIONS

In this paper, we proposed a new holistic viewpoint of instantly decodable network coding (IDNC) schemes that simultaneously takes into account both the overall completion time (OCT) and decoding delay and improves the balance between these two performance metrics for broadcast transmission over erasure channels with a wide range of memory conditions. We formulated the IDNC packet selection for such systems using an SSP technique. However, since solving the SSP problem in the proposed scheme is computationally complex, we further proposed two different heuristic algorithms that each improves this balance between the OCT and decoding delay for a specific range of channel memory conditions. Furthermore, it was shown that the proposed scheme offers

Fig. 7: (a) OCT versus decoding delay, and (b) variance of the decoding delay for different number of packets N for channel memory $\mu = 0.6$ and $M = 30$ receivers.

a more uniform (fair) decoding delay experience across all receivers. Extensive simulations were conducted to assess the performance of the proposed algorithms compared to the best known existing algorithms in the literature. The simulation results show that our proposed algorithms achieve an improved balance between the OCT and decoding delay.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that greatly contributed to improving the quality of the paper.

REFERENCES

- [1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Trans. Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [3] C. Fragouli, J. Widmer, and J. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Networking*, vol. 16, no. 2, pp. 450–463, 2008.
- [4] C. Fragouli, D. Lun, M. Médard, and P. Pakzad, "On feedback for network coding," in *Proc. 2007 Conference on Information Sciences and Systems*, pp. 248–252.
- [5] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. 2008 Workshop on Network Coding, Theory and Applications*, pp. 1–6.
- [6] A. Eryilmaz, A. Ozdaglar, and Médard, "On the delay performance gains for network coding," in *Proc. 2006 Conference on Information Sciences and Systems*, pp. 864–870.
- [7] R. Costa, D. Munaretto, J. Widmer, and J. Barros, "Informed network coding for minimum decoding delay," in *Proc. 2008 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 80–91.
- [8] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. 2009 IEEE International Symposium on Information Theory*, pp. 844–848.
- [9] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. 2009 IEEE Conference on Computer Communications*, pp. 208–216.
- [10] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. 2009 IEEE Conference on Computer Communications*, pp. 190–198.
- [11] J. K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. 2009 Workshop on Network Coding, Theory, and Applications*, pp. 1–6.
- [12] A. Yazdi, S. Sorour, S. Valaee, and R. Kim, "Optimum network coding for delay sensitive applications in WiMAX unicast," in *Proc. 2009 IEEE Conference on Computer Communications*, pp. 2576–2580.
- [13] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. Wireless Commun. and Networking*, pp. 1–14, Jan. 2010.
- [14] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. 2010 IEEE Global Telecommunications Conference*, pp. 1–5.
- [15] —, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. 2010 IEEE International Conference on Communications*, pp. 1–5.
- [16] B. Swapna, A. Eryilmaz, and N. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," in *Proc. 2010 International Symposium on Network Coding*, pp. 1–6.
- [17] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Trans. Networking*, vol. 19, no. 3, pp. 869–878, Jun. 2011.
- [18] M. Nistor, D. E. Lucani, T. T. V. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1084–1093, May 2011.
- [19] P. Sadeghi and M. Yu, "Instantly decodable versus random linear network coding: a comparative framework for throughput and decoding delay performance." Available: <http://arxiv.org/abs/1208.2387>
- [20] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [21] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.
- [22] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *Proc. 2007 Packet Video Workshop*, pp. 1–9.
- [23] X. Li, C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, May 2011.
- [24] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914–925, Feb. 2009.
- [25] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes." Available: <http://arxiv.org/abs/1201.4768>
- [26] X. Li, C. Wang, and X. Lin, "Throughput and delay analysis of uncoded and coded wireless broadcast with hard deadline constraints," in *Proc. 2010 IEEE Conference on Computer Communications Workshops*.
- [27] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proc. 2009 Workshop on Network Coding, Theory and Applications*, pp. 1–6.
- [28] S. Wang, C. Gong, X. Wang, and M. Liang, "Instantly decodable network coding schemes for in-order progressive retransmission," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1069–1072, Jun. 2013.

- [29] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc. 2013 Workshop on Network Coding, Theory and Applications*.
- [30] J. K. Sundararajan, D. Shah, and M. Médard, "Feedback-based online network coding," Available: [url{http://arxiv.org/abs/0904.1730}](http://arxiv.org/abs/0904.1730)
- [31] P. Sadeghi, R. A. Kennedy, P. Rapajic, and R. Shams, "Finite-state Markov modeling of fading channels: a survey of principles and applications," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 57–80, Sep. 2008.
- [32] M. S. Karim and P. Sadeghi, "Decoding delay reduction in broadcast erasure channels with memory for network coding," in *Proc. 2012 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5.
- [33] S. Sorour, N. Aboutorab, P. Sadeghi, M. S. Karim, T. Al-Naffouri, and M. Alouini, "Delay reduction in persistent erasure channels for generalized instantly decodable network coding," in *Proc. 2013 IEEE Vehicular Technology Conference*, pp. 1–5.
- [34] D. Nguyen and T. Nguyen, "Network coding-based wireless media transmission using POMDP," in *Proc. 2009 Packet Video Workshop*, pp. 1–9.

data storage systems, and signal processing.

Neda Aboutorab (S09-M13) received the B.E. and M.E. degrees in electrical engineering from Iran University of Science and Technology and Amirkabir University of Technology, Tehran, Iran, in 2005 and 2008, respectively, and the Ph.D. degree in electrical engineering from The University of Sydney, Sydney, Australia, in 2013. She is currently a Postdoctoral Research Fellow at the Research School of Engineering, The Australian National University, Canberra, Australia. Her research interests include network coding, wireless communications,

Engineering, The Australian National University, Canberra, Australia. She has visited various research institutes, including the Institute for Communications Engineering, Technical University of Munich in 2008 and MIT in 2009 and 2013.

Dr. Sadeghi has co-authored more than 100 refereed journal or conference papers and a book on Hilbert Space Methods in Signal Processing, published by Cambridge University Press in 2013. She is a Chief Investigator in a number of Australian Research Council Discovery and Linkage Projects. In 2003 and 2005, she received two IEEE Region 10 student paper awards for her research in the information theory of time-varying fading channels. Her research interests are mainly in the areas of network coding, wireless communications systems and signal processing.

Parastoo Sadeghi (S02-M06-SM07) received the B.E. and M.E. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from The University of New South Wales, Sydney, Australia, in 2006. From 1997 to 2002, she worked as a Research Engineer and then as a Senior Research Engineer at Iran Communication Industries (ICI) in Tehran, Iran and at Deqx (formerly known as Clarity Eq) in Sydney, Australia.

She is currently a Fellow at the Research School of Engineering, The Australian National University, Canberra, Australia. She has visited various research institutes, including the Institute for Communications Engineering, Technical University of Munich in 2008 and MIT in 2009 and 2013.

Laboratory (WIRLab), Department of Electrical and Computer Engineering, University of Toronto, Canada. His research interests include network coding, indoor localization, cooperative data exchange, sensor, vehicular and cognitive radio networks, distributed cloud storage and smart networks.

Sameh Sorour (S98-M11) is an Assistant Professor at the Electrical Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM). He received his B.Sc. and M.Sc. degrees in Electrical Engineering from Alexandria University, Egypt, in 2002 and 2006, respectively. In 2002, he joined the Department of Electrical Engineering, Alexandria University, where he was a Teaching and Research Assistant for three years and was promoted to Assistant Lecturer in 2006. In 2011, he obtained his Ph.D. degree from the Wireless and Internet Research