On Improving the Balance between the Completion Time and Decoding Delay in Instantly Decodable Network Coded Systems

Neda Aboutorab†, Parastoo Sadeghi†, and Sameh Sorour∗
Email: †{neda.aboutorab, parastoo.sadeghi}@anu.edu.au, ∗samehsorour@kfupm.edu.sa

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, and proposes two new algorithms that improve the balance between the completion time and decoding delay of broadcasting a block of packets. We first formulate the IDNC packet selection problem that provides joint control of the completion time and decoding delay as a statistical shortest path (SSP) problem. However, since finding the optimal 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 heuristic packet selection algorithms that can efficiently improve the balance between the completion time and decoding delay 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], [2], [3] 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. For example, NC has been considered for delivering high data rate multimedia broadcast or multicast services (MBMS) [4], [5], [6], [7]. In addition to being high data rate in nature, such applications also often have strict delay requirements. However, the higher throughput offered by NC does not necessarily translate into faster delivery of information to the application [8], [9]. 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 [8], [10], [9], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [6], [21], [22].

An important example that illustrates the tension between throughput and delay is random linear network coding (RLNC) [20], [21], [23] 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 (block completion time) among block-based NC schemes [8], [10], [20]. 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 NC systems, 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. A decoding delay occurs at a receiver when it is not targeted in an IDNC transmission. That is, it receives a packet that contains either no or more than one desired packets of that receiver. Compared to RLNC, IDNC in broadcast erasure channels can have a lower throughput. In other words, IDNC incurs a generally higher completion time for the broadcast of the same number of $N$ packets. However, it can provide a faster delivery of uncoded packets to the application layer, as required for MBMS. Therefore, similar tension between throughput and delay can also be observed in IDNC.

Inspired by the low-complexity XOR-based encoding and decoding process of IDNC and its potential application in MBMS and unicast settings [5], [9], [17], [18], [19], [24], [25], [26], in this paper we are interested in understanding the interplay between its throughput and delay over broadcast erasure channels and proposing novel IDNC schemes that offer a better control of these performance metrics.

The problem of maximizing the throughput for a deadline-constrained video-streaming scenario is considered in [5], where each packet has a delivery deadline and has to be decoded before the deadline, otherwise it is expired. In this paper, however, we consider a block-based transmission, where all the packets in the block have to be received by all the receivers and there is no packet deadline. Furthermore, in this paper, no new packet arrival is considered in the system while the transmission of a block is in progress. In addition, this study is applicable where partial decoding is beneficial and can result in lower delays irrespective of the order in which packets are being decoded. Examples of such applications can be found in sensor or emergency networks and multiple-description source coded systems [27], in which every decoded packet brings new information to the destination, irrespective of its order.

In this context, the closest works to ours are [18], [7] and [19]. In particular, the authors in [18] 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 [19], [7], 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 [18], [19], [7] 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 [18], [7] and [19] 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 the 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 [18], [7] and [19]. 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 [28], where the packet erasures occur in bursts, due to deep fading and shadowing. By following the proposed channel models in [17], [28], [29], [30], we model the bursts of erasures (i.e. the memory of the channel) by a simple two-state Gilbert-Elliott channel (GEC) model and propose two algorithms that can offer an improved balance between the 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 optimal packet selection that provides an improved balance between the completion time and decoding delay for broadcast transmission over memoryless channels as an SSP problem. However, since finding the optimal packet selection in the proposed SSP scheme is computationally complex, we use the SSP formulation and its geometric structure to find some guidelines that can be used to propose a new heuristic packet selection algorithm that efficiently improves the balance between the 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 and improve the balance between the completion time and decoding delay by selecting the packet combinations more effectively based on the channel memory conditions compared to the algorithms that are ignorant to the channel memory. Finally, by taking into account both the number of missing packets and the decoding delay of the receivers, 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 is 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 to find the properties of the optimal packet selection policy. 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 encoded 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) \( F = [f_{ij}], \forall i \in M, j \in N \) as:

\[
f_{ij} = \begin{cases} 
0 & j \in H_i \\
1 & j \in W_i 
\end{cases}
\]  

(1)

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

\[
F = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\]  

(2)

In this example, \( f_{11} = 1 \) denotes that packet 1 is missed at receiver 1, and \( f_{23} = 0 \) 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 \( W_i \) and \( H_i \), 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 [17, 18, 19, 7, 25, 26].

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 \( W_i \).

2) Instantly decodable packet: A packet is instantly decodable for receiver \( i \) if it contains only one source packet from \( W_i \). 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 \( W_i \).

**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 form receiver 3’s Wants set.

We define the completion time and decoding delay similar to [9, 18, 17, 19, 7, 26] as follows:

**Definition 1:** Individual completion time (ICT) of receiver \( i \), denoted by \( T^i \), 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, \( T^i \) will be equal to the size of its Wants set at the start of the recovery transmission phase, i.e. \( T^i = |W^i| \).

**Definition 2:** Overall completion time (OCT), denoted by \( T \), 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, \( D^i_t = \sum_{l=1}^{t} d^i_l \).

III. IDNC PACKET GENERATION

In this paper, we adopt IDNC [18, 19] as our NC transmission scheme. IDNC allows the sender to transmit a coded packet that includes at most one source packet from the Wants sets of the targeted receivers (either an appropriately selected subset or if possible all receivers). Thus, at the targeted receivers, the 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, 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 in the form of a graph model, which was first used in the context of IDNC in [18], [19]. Then, we will briefly review the packet selection schemes in [19] and [18] that were used to separately minimize IDNC’s OCT and decoding delay, respectively.

As presented in [18], [19], the IDNC graph \( G(V, E) \) is constructed by first inducing a vertex \( v_{ij} \in V \) for each packet \( j \in W_i, \forall i \in M \). In other words, any vertex \( v_{ij} \) represents a wanted packet \( j \) for receiver \( i \). Two vertices \( v_{ij} \) and \( v_{ik} \) in \( V \) are connected by an edge \( E \) 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:** \( j \in H_k \) and \( l \in H_i \) \( \Rightarrow \) The wanted packet corresponding to each vertex \( v_{ik} \) 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 \( G \) (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 \( G \). Assuming that \( \kappa \) is the selected maximal clique in \( G \), the targeted receivers of this clique are represented by \( T(\kappa) \).

The problem of minimizing the OCT of the IDNC scheme for broadcast erasure channels has been studied in [19] where it is shown that the expected ICT for receiver \( i \), denoted by \( \tau_i \), if addressed in all future transmissions, can be expressed as \( \tau_i = \frac{\delta_i}{1-p_i} \). Having the expected ICT of all receivers calculated, it is shown in [19] that an efficient policy for reducing the OCT should select maximal cliques that include the maximum number of vertices belonging to receivers having the largest \( \tau_i \). In order to simplify such maximal cliques selection in the IDNC graph \( G \), the authors in [19] proposed a maximum weight vertex search algorithm, where the weights of vertices in \( G \) reflect 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 \( G \) such that:

\[
a_{ij,kl} = \begin{cases} 
1 & \text{if } v_{ij} \text{ is connected to } v_{kl} \text{ in } G, \\
0 & \text{otherwise.}
\end{cases}
\]

Given the adjacency indicator, the weighted degree \( \Delta_{ij} \) of vertex \( v_{ij} \) in [19] is defined as \( \Delta_{ij} = \sum_{\forall v_{kl} \in G} a_{ij,kl} \tau_k \). 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 \( \tau_i \) value and is connected to a large number of vertices having large \( \tau_k \) values.

The problem of minimizing the decoding delay of IDNC scheme for broadcast erasure channels has been studied in [18] where it is shown that an efficient policy for reducing decoding delay is selecting maximal cliques that include the maximum 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 \( v_{ij} \) in [18] is defined as \( w_{ij} = \Delta_{ij}(1 - p_i) \), where \( \Delta_{ij} \) reflects the connection of vertex \( v_{ij} \) to vertices having large reception probabilities and is defined in [18] as \( \Delta_{ij} = \sum_{\forall v_{kl} \in G} a_{ij,kl}(1 - p_k) \).

**Example 3:** Let us again consider the SFM in (2). By using the technique in [19] 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 [18] is adopted to minimize 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. In this example, it can be easily seen that minimizing the OCT on its own may result in an increased decoding delay and also minimizing the decoding delay alone may result in an increased OCT of the transmission.

Unlike [19] and [18], 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 offers an improved balance between the OCT and decoding delay performances, and at the same time provides a more uniform decoding delay experience across all receivers for the broadcast of \( N \) packets. 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 decoding delay that the respective receiver experiences. Furthermore, we note that the OCT of the IDNC transmission is equal to the maximum of ICTs. We will use this relationship between the OCT and decoding delay to design an IDNC scheme that provides a balance between these two performance metrics. The proposed scheme is then solved as an SSP problem.

Here, we first define \( \mathbf{W}^s = [W^s_1, ..., W^s_M] \) and \( \mathbf{H}^s = [H^s_1, ..., H^s_M] \) as the Wants and Has vectors, such that \( W^s_i \) and \( H^s_i \) are the cardinalities of Wants and Has sets at the start of recovery phase, \( W^s_i \) and \( H^s_i \), respectively. Furthermore, \( \mathbf{D}^f = [D^f_1, ..., D^f_M] \) is defined as the final accumulative decoding delay vector, where \( D^f_i \) 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, \( W^s_i \), 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. \( \max_{i \in \mathcal{M}} \{W^s_i\} \)). Furthermore \( D^f_i = 0 \), \( \forall i \in \mathcal{M} \).

However, since it is not always possible to target all the receivers with non-empty wants sets in every single 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 time-slots that they were not targeted). Therefore, we can write the
ICT of receiver \( i \), denoted by \( T^f_i \), as
\[
T^f_i = W^s_i + D^f_i, \quad i \in \mathcal{M}
\]  
(4)

As shown in (4), the ICT of each receiver depends on the size of its initial Wants set, \( W^s_i \), and the final accumulative decoding delay it experiences, \( D^f_i \). 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
\[
T^f = \max_{i \in \mathcal{M}} T^f_i = \max_{i \in \mathcal{M}} \{ W^s_i + D^f_i \}
\]  
(5)

It is worth noting that based on (4), minimizing the 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. Therefore, the problem of providing a balance between the decoding delay and OCT can be translated into balancing between \( \min_{i \in \mathcal{M}} T^f_i \) and \( \min \max_{i \in \mathcal{M}} T^f_i \) of the receivers.

In the next section, we will show that the packet selection problem that offers such balance between the OCT and decoding delay of the receivers for the IDNC can be formulated in the form of 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 [7] in order to select the packet combinations that result in minimum completion time. In SSP problem, different possible situations that may arise during recovery transmission phase 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 all the receivers receive all their missing packets. 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. \( D(s) = [D_1(s), ..., D_M(s)] \) where \( D_i(s) = D^f_i, \forall i \in \mathcal{M} \); depends on the taken actions in all states prior to reaching the absorbing state.

1. **State Space** \( \mathcal{S} \): Each state \( s \) can be characterized by its Has, Wants and the accumulative decoding delay vectors, \( \mathbf{H}(s) = [H_1(s), ..., H_M(s)] \), \( \mathbf{W}(s) = [W_1(s), ..., W_M(s)] \) and \( D(s) = [D_1(s), ..., D_M(s)] \), respectively.

The values of \( \mathbf{W}(s) \), \( \mathbf{H}(s) \) and \( D(s) \) at the starting state of the recovery transmission phase, \( s_s \), are represented by \( \mathbf{W}(s_s) = [W_1(s_s), ..., W_M(s_s)] \), \( \mathbf{H}(s_s) = [H_1(s_s), ..., H_M(s_s)] \) and \( D(s_s) = [0, ..., 0] \), respectively, where \( W_i(s_s) = W^s_i \) and \( H_i(s_s) = H^s_i, \forall i \in \mathcal{M} \). Furthermore, we define the absorbing state, \( s_a \), as the state in which all the receivers receive all their missing packets.

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

3. **State-Action Transitions Probabilities:** Considering the fact that each state \( s \) can be efficiently represented by the Wants sets and accumulative decoding delays of all receivers, here, we further define the **state value** \( U_i(s) \) for receiver \( i \) as
\[
U_i(s) = W_i(s) + D_i(s)
\]  
(8)

In a more general framework, this equation can be written as
\[
U_i(s) = \lambda W_i(s) + (1 - \lambda) D_i(s),
\]  
(9)

where the weight \( \lambda \) can be designed for more control over OCT or decoding delay according to the system requirements. In the rest of this paper, we assign equal weights to \( W_i(s) \) and \( D_i(s) \), and consider the state value of receiver \( i \) to be of the form \( U_i(s) = W_i(s) + D_i(s) \), except stated otherwise.

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

To define \( P_a(s, s') \), here, we first introduce the following three sets:

\[
\mathcal{X} = \{ i \in T(\kappa(s)) \mid U_i(s') < U_i(s) \}
\]  
(10)

\[
\mathcal{Y} = \{ i \in \mathcal{M}_w(s) \mid T(\kappa(s)) \mid U_i(s') > U_i(s) \}
\]  
(11)
The state representation and action space for this SFM are sets at state \( s \) where
\[ M \]
packet receivers in the maximal clique \( \kappa(s) \) states given that action
\[ \kappa \]
ions for action \( W \) and their accumulative decoding delays remain unchanged (i.e. \( \{0,1,2\} \), associated with action \( a \) on receiver \( i \) that can be expressed as follows:
\[ \text{Costs:} \]
- \( C_i(s,a,s') = 0 \) 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. \( i \in T(a) \), the cost will be
\[ c_i(s,a,s'|i \in T(a)) = \begin{cases} 0 \quad &\text{with prob.} \; (1-p_i) \\ 1 \quad &\text{with prob.} \; p_i \end{cases} \]  
(15)

Thus, the expected cost given receiver \( i \) is targeted by action \( a \) can be calculated as
\[ \bar{c}_i(s,a|i \in T(a)) = 0 \times (1-p_i) + 1 \times p_i = p_i \]  
(16)

However, if receiver \( i \) is not targeted by action \( a \), i.e. \( i \notin T(a) \), the cost will be
\[ c_i(s,a,s'|i \notin T(a)) = \begin{cases} 2 \quad &\text{with prob.} \; p_i \\ 1 \quad &\text{with prob.} \; (1-p_i) \end{cases} \]  
(17)

Thus, the expected cost given receiver \( i \) is not targeted by action \( a \) can be calculated as
\[ \bar{c}_i(s,a|i \notin T(a)) = 1 \times p_i + 2 \times (1-p_i) = 2 - p_i \]  
(18)

The total expected cost of action \( a \) over all the receivers in \( M_w(s) \) can thus be defined as
\[ \bar{c}(s,a) = \sum_{i \in M_w(s)} \bar{c}_i(s,a|i \in T(a)) + \sum_{i \in M_w(s)} \bar{c}_i(s,a|i \notin T(a)) \]
\[ = \sum_{i \in T(a)} p_i + \sum_{i \in (M_w(s) \setminus T(a))} (2-p_i) \]  
(19)
5) Optimal Policy: The optimal policy as presented in Section IV-A can be expressed as

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

\[
= \arg \min_{a \in A(s)} \{ \bar{c}(s, a) + \mathbb{E}_a[V_{\pi^*}(s')] \} \tag{20}
\]

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 [31]. Furthermore, there is no closed-form solution to this problem. Thus, instead of solving the SSP problem formulated in (20), we can study its properties and structure to draw the characteristics of the optimal policy. To this end, we will study the geometric structure of the SSP solution in the context of the proposed IDNC scheme. 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 \( U(s) = [U_1(s), ..., U_M(s)] \), 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 \( \Delta = [\delta_1, ..., \delta_M] \) in this space all the states that have the state vectors \( U(s) = [U_1(s), ..., U_M(s)] \) 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 \( W_i(s) = 0, \forall i \in M \). Under the special scenario where the accumulative decoding delays are zero, i.e. \( D_i(s) = 0, \forall i \in M \), 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 \( \Delta = U(s) \) to another point \( \Delta' = U(s') \) which is a vertex in the hypercube \( \Gamma(s) \) defined as:

\[
\Gamma(s) = \{ \Delta | U_i(s') - U_i(s) \in \{-1, 0, 1\}, \forall i \in M_w \} \tag{21}
\]

In other words, \( \Gamma(s) \) is the hypercube of side length 1, in which \( U(s) \) and \( U(s') \) 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.

Case 1: 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, \( U(s) \) and \( U(s') \) are always two diagonal corners in the hypercube \( \Gamma(s) \), i.e. \( \Gamma(s) = \{ \Delta | U_i(s') - U_i(s) \in \{-1, 1\}, \forall i \in M_w \} \).

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 \( c_i(s, a) = 0, \forall i \in M_w \), 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 (14). 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 do not always exist in most states, 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 (14), 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 Guidelines]** 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 having higher priorities for receivers with larger values of $U_i(s)$ 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 also be translated into improving the decoding delay fairness among the receivers while minimizing the OCT of the system. 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). It is worth noting that this greedy scheme does not necessarily result in a globally optimal policy stated in (20).

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 $U(s)$. According to these facts and the above discussion in Subcase 2, 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 $\tilde{U}(s) = [\tilde{U}_1(s), ..., \tilde{U}_M(s)]$, where

$$\tilde{U}_i(s) = \tilde{W}_i(s) + D_i(s) = \frac{W_i(s)}{1 - p_i} + D_i(s) \quad (22)$$

Based on this new vector definition, we can re-define our space such that the points $\Delta$ are identified by the coordinates of the vectors $\tilde{U}(s)$ instead of $U(s), \forall s \in S$. 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:

$$\Gamma'(s) = \{ \Delta' | \tilde{U}_i(s') - \tilde{U}_i(s) \in \{-\frac{1}{1 - p_i}, 0, 1\}, \forall i \in M \} \quad (23)$$

In the next section, by 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 the clique 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 $(\tilde{U}_k(s))^2$ values of their inducing receivers, but also their adjacency to the vertices with high $(\tilde{U}_k(s))^2$.

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

$$\Theta_{ij}(s) = \sum_{\forall v_{ij} \in G(s)} a_{ij,kl}(\tilde{U}_k(s))^2 \quad (24)$$

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:

$$w_{ij}(s) = (\tilde{U}_i(s))^2 \Theta_{ij}(s) \quad (25)$$

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 $\kappa^*$. Note that at first, $\kappa^*$ 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 \( \kappa^* \), denoted by \( G_{\kappa^*}(s) \), 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 \( \kappa^* \). 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 \( G(s) \) based on \( F(s) \).
2) While \( G(\kappa^*(s)) \neq \emptyset \) do
   Compute \( w_{ij}(s), \forall v_{ij} \in G(\kappa^*(s)) \) using (3), (24) and (25).
   Select \( v^* = \arg \max_{v_{ij} \in G(\kappa^*(s))} \{ w_{kl}(s) \} \).
   Set \( \kappa^*(s) \leftarrow \kappa^*(s) \cup v^* \).
   Update subgraph \( G(\kappa^*(s)) \).

VII. Heuristic Packet Selection Algorithm for Erasure Channels with Memory

In this section, our goal is to extend our proposed MWVS scheme to the coded transmissions in erasure channels with memory. To model erasure channels with memory, we employ the well-known Gilbert-Elliott channel (GEC) [28] 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 to the bad state is \( b \triangleq Pr(G \rightarrow B) \) and the probability of moving from the bad state to the good state \( G \triangleq Pr(B \rightarrow G) \). Steady-state probabilities are derived as \( P_G \triangleq Pr(C_i = G) = \frac{a}{b+g} \) 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 [28], 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 \( \mu \) means a channel with little memory and a large \( \mu \) 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 the 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 (22) as

\[
\tilde{U}_i(s) = \frac{W_i(s)}{Pr(C_i \rightarrow G)} + D_i(s)
\]  

In other words, the weight of each vertex in (25) can now be recalculated based on the conditional reception probability of its inducing receiver, given its previous state, as

\[
w_{ij}(s) = (\tilde{U}_i(s))^2 + W_i(s) \frac{1}{Pr(C_i \rightarrow G)} + D_i(s)^2 \Theta_{ij}(s)
\]

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 [30]. The proposed algorithm comprises two different layers of subgraphs. The first layer of subgraph, \( G_0(s) \subset G(s) \), consists of vertices of GCRs. In the first step, the MWVS algorithm is applied on the subgraph \( G_0(s) \), and \( \kappa_0^*(s) \) is obtained. Then, in the second step, the algorithm finds \( \kappa_1^*(s) \) by applying the MWVS algorithm another time on the second layer of subgraph, \( G_0(s) \), consisting of BCRs that are adjacent to all the vertices of the chosen clique \( \kappa_0^*(s) \). Thus, the final clique can be obtained by the union of the cliques from the two layers as \( \kappa^*(s) = \kappa_0^*(s) \cup \kappa_1^*(s) \). The steps of MWVS-Layered algorithm are summarized in Algorithm 2.

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 [19], [18], [30] over
Algorithm 2 Proposed MWVS-Layered Algorithm

1) Initialize $\kappa_g^*(s) = \emptyset$ and $\kappa_b^*(s) = \emptyset$
   Construct $G(s)$ based on $F(s)$.
   Form $G_g(s)$ and $G_b(s)$ according to the channels’ previous states $C_i$, $\forall i \in \mathcal{M}$.
2) While $G_g(\kappa_g^*(s)) \neq \emptyset$ do
   Compute $w_{ij}(s), \forall v_{ij} \in G_g(\kappa_g^*(s))$ using (3), (24) and (27).
   Select $v^* = \arg \max_{w_{ij} \in G_g(\kappa_g^*(s))} \{w_{ij}(s)\}$.
   Set $\kappa_g^*(s) \leftarrow \kappa_g^*(s) \cup v^*$.
   Update subgraphs $G_g(\kappa_g^*(s))$ and $G_b(\kappa_b^*(s))$.
3) While $G_b(\kappa_b^*(s)) \neq \emptyset$ do
   Compute $w_{ij}(s), \forall v_{ij} \in G_b(\kappa_b^*(s))$ using (27).
   Select $v^* = \arg \max_{w_{ij} \in G_b(\kappa_b^*(s))} \{w_{ij}(s)\}$.
   Set $\kappa_b^*(s) \leftarrow \kappa_b^*(s) \cup v^*$.
   Update subgraph $G_b(\kappa_b^*(s))$.
4) $\kappa^*(s) = \kappa_g^*(s) \cup \kappa_b^*(s)$

...a wide range of channel memory conditions. Furthermore, 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.

We start with our simulation results for memoryless erasure channels and compare the performance of our proposed MWVS algorithm with the schemes in [19] and [18], denoted by “Min-OCT” and “Min-DD”, respectively. Furthermore, we have simulated the proposed scheme for $\lambda = 0$ and 1, denoted by “MWVS ($\lambda = 0$)” and “MWVS ($\lambda = 1$)”, respectively. $\lambda = 0$ corresponds to the case that the objective of the proposed scheme is to reduce the accumulative decoding delay and $\lambda = 1$ corresponds to the case where the objective of the proposed scheme is to reduce the OCT of the system in each time slot. The simulation results of the proposed MWVS algorithm when equal weights are assigned to $W_i(s)$ and $D_i(s)$, as in (8), are denoted by “MWVS”.

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 is presented in Figure 3(b). From these figures, we first observe that the Min-OCT algorithm in [19] that achieves the minimum OCT among the IDNC schemes in Figures 3(a) and 3(b), results in the worst decoding delay performance, and the Min-DD algorithm in [18] that achieves the minimum decoding delay performance, results in the worst OCT performance. 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 [19]. 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 [18]. 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 the Min-DD algorithm in [18] aims
to reduce the decoding delay in each time-slot (defined in Definition 3).

Figure 4 illustrates the variance of the decoding delay 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 the variance of the decoding delay. This can be translated into a better fairness in the decoding delay experienced by different receivers.

For erasure channels with memory, the full graph search and the layered graph search algorithms proposed in [30] 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 [19] to become channel memory aware by replacing the probability of successful reception at receiver $i$ with $Pr(C_i \to G)$. 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, 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 [18] giving higher priorities to the receivers with higher probabilities of successful reception improves the decoding 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.

Figure 7 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.

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 optimal 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 a more uniform 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.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” *IEEE Transactions on Information 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 Transactions on 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 Transactions on Networking*, vol. 16, no. 2, pp. 450–463, 2008.
[4] D. Nguyen, T. Nguyen, and X. Yang, “Multimedia wireless transmission with network coding,” in *Proc. Packet Video Workshop (PV)*, 2007, pp. 1–9.
[5] 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 Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1094–1105, May 2011.
[6] S. Sorour and S. Valaee, “An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 869–878, Jun. 2011.
[7] ——, Completion delay minimization for instantly decodable network codes. [Online]. Available: http://arxiv.org/abs/1201.4768
[8] C. Fragouli, D. Lun, M. Médard, and P. Pakzad, “On feedback for network coding,” in *Proc. Annual Conference on Information Sciences and Systems (CISS)*, 2007, pp. 248–252.
[9] L. Keller, E. Drinea, and C. Fragouli, “Online broadcasting with network coding,” in *Proc. Workshop on Network Coding, Theory and Applications (NETCOD)*, 2008, pp. 1–6.
[10] A. Eryilmaz, A. Ozdaglar, and Médard, “On the delay performance gains for network coding,” in *Proc. 40th Annual Conference on Information Sciences and Systems (CISS)*, 2006, pp. 864–870.
[11] R. Costa, D. Munaretto, J. Widmer, and J. Barros, “Informed network coding for minimum decoding delay,” in *Proc. IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2008, pp. 80–91.
[12] E. Drinea, C. Fragouli, and L. Keller, “Delay with network coding and feedback,” in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2009, pp. 844–848.
[13] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, “Effective delay control in online network coding,” in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2009, pp. 208–216.
[14] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, “Minimizing delay for multicast-streaming in wireless networks with network coding,” in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2009, pp. 190–198.
[15] J. K. Sundararajan, P. Sadeghi, and M. Médard, “A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay,” in Proc. Workshop on Network Coding, Theory, and Applications (NETCOD), 2009, pp. 1–6.

[16] A. Yazdi, S. Sorour, S. Valaee, and R. Kim, “Optimum network coding for delay sensitive applications in WiMAX unicast,” in Proc. IEEE Conference on Computer Communications (INFOCOM), Apr. 2009, pp. 2576–2580.

[17] P. Sadeghi, R. Shams, and D. Traskov, “An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels,” EURASIP Journal on Wireless Communications and Networking, pp. 1–14, Jan. 2010.

[18] S. Sorour and S. Valaee, “Minimum broadcast decoding delay for generalized instantly decodable network coding,” in Proc. IEEE Global Telecommunications Conference (GLOBECOM), Dec. 2010, pp. 1–5.

[19] ——, “On minimizing broadcast completion delay for instantly decodable network coding,” in Proc. IEEE International Conference on Communications (ICC), May 2010, pp. 1–5.

[20] B. Swapna, A. Eryilmaz, and N. Shroff, “Throughput-delay analysis of random linear network coding for wireless broadcasting,” in Proc. International Symposium on Network Coding (NETCOD), Jun. 2010, pp. 1–6.

[21] 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 Journal on Selected Areas in Communications, vol. 29, no. 5, pp. 1084–1093, May 2011.

[22] P. Sadeghi and M. Yu. Instantly decodable versus random linear network coding: A comparative framework for throughput and decoding delay performance. [Online]. Available: http://arxiv.org/abs/1208.2387

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

[24] P. Sadeghi, D. Traskov, and R. Koetter, “Adaptive network coding for broadcast channels,” in Proc. Workshop on Network Coding, Theory and Applications (NETCOD), 2009, pp. 1–6.

[25] S. Wang, C. Gong, X. Wang, and M. Liang, “Instantly decodable network coding schemes for in-order progressive retransmission,” IEEE Communications Letters, vol. 17, no. 6, pp. 1069–1072, Jun. 2013.

[26] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, “Instantly Decodable Network Codes for Real-Time Applications,” in Proc. Workshop on Network Coding, Theory and Applications (NETCOD), 2013.

[27] Y. Wang, A. R. Reibman, and S. Lin, “Multiple description coding for video delivery,” Proceedings of IEEE, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[28] 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 Processing Magazine, vol. 25, no. 5, pp. 57–80, Sep. 2008.

[29] M. S. Karim and P. Sadeghi, “Decoding delay reduction in broadcast erasure channels with memory for network coding,” in Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sep. 2012, pp. 1–5.

[30] S. Sorour, N. Aboutorab, P. Sadeghi, M. S. Karim, T. Al-Naffouri, and M. Aiouini, “Delay reduction in persistent erasure channels for generalized instantly decodable network coding,” in Proc. IEEE Vehicular Technology Conference (VTC), Jun. 2013, pp. 1–5.

[31] D. Nguyen and T. Nguyen, “Network coding-based wireless media transmission using POMDP,” in Proc. Packet Video Workshop (PV), 2009, pp. 1–9.