Decoding Delay Reduction in Network Coded Cooperative Systems with Intermittent Status Update

Mohammad S. Karim, Neda Aboutorab, Ali A. Nasir and Parastoo Sadeghi
Research School of Engineering, The Australian National University, Canberra, Australia
E-mail: {mohammad.karim, neda.aboutorab, ali.nasir, parastoo.sadeghi}@anu.edu.au

***Abstract*—In this paper, we study the problem of decoding delay reduction for instantly decodable network coding (IDNC) in broadcast cooperative systems, where a group of closely located clients cooperate with each other to obtain their missing packets. In such cooperative systems, one of the clients (referred to as the *leader*) decides the transmitting client and the packet combination for each transmission. We consider intermittent system status update (SSU) at the leader such that all other clients feed back their packet reception status to the leader after several cooperative transmissions. We first introduce an intermittent local IDNC (IL-IDNC) graph to represent all potential packet combinations for a transmitting client. We then formulate the joint client and packet selection problem that results in the minimum expected decoding delay in each cooperative transmission as a maximum weight clique problem over all the IL-IDNC graphs. Since solving the formulated problem is computationally complex, we propose a heuristic algorithm to select the transmitting client and the packet combination that can reduce the decoding delay. Simulation results show that the proposed heuristic algorithm can achieve a tolerable degradation compared to the full SSU performance while using a smaller number of SSUs.**

***Index Terms*—Instantly Decodable Network Coding, Decoding Delay, Cooperative Systems, Wireless Networks.**

I. Introduction

Imagine a group of geographically close clients that are interested in obtaining a common set of packets and each client initially holds a subset of these packets. The clients cooperate with each other to obtain their missing packets and achieve a common objective in the system. An example of such cooperative systems is a wireless sensor network. At the beginning, a central unit broadcasts a set of packets (i.e. commands) to a group of ground sensors in its reception zone. However, due to erasures in wireless channels, each sensor receives a subset of packets. When the sensors collectively receive all the packets, they can cooperate with each other to obtain their missing packets. Such cooperative system has recently drawn significant attention [1]–[6] due to its numerous advantages, such as load reduction of the central unit, and fast and reliable delivery of the packets to the clients through short-range communication channels [4].

In the considered sensor network, the sensors can execute a new command upon receiving a new packet irrespective of its order and fast command execution at the sensors is crucial. In such time-critical and order-insensitive applications [7], if a received packet at a client does not bring new information or cannot be immediately decoded, the client experiences one unit *decoding delay*. To minimize the decoding delay, an attractive strategy is to employ instantly decodable network coding (IDNC) [6]–[11]. IDNC exploits the diversity of received and lost packets at different clients to generate packet combinations that are instantly decodable at the clients. Moreover, IDNC uses simple XOR-based encoding and decoding operations and does not require decoding buffers at the clients to store non-instantly decodable packets for future decoding possibilities [7], [11].

The problem of minimizing the decoding delay of IDNC in the point-to-multipoint (PMP) networks (such as cellular and Wi-Fi) was studied in [7], [8] for full feedback scenario and then extended to intermittent feedback scenario in [10], [11]. In IDNC systems, the sender needs the packet reception status of the clients to make efficient coding decisions. Intermittent feedback from the clients creates uncertainties at the sender about their packet reception status and thus, affects the performance of IDNC as shown in [10], [11].

Unlike the PMP works, the authors in [3], [4] considered the network coded cooperative systems and addressed the problem of reducing the number of cooperative transmissions. Moreover, in [6], the authors considered the problem of reducing the decoding delay of IDNC in cooperative systems and formulated the joint client and packet selection problem that results in the minimum decoding delay as a maximum weight clique problem over all clients' local IDNC graphs. This formulated problem was solved at all clients so that the clients can separately decide the transmitting client and the packet combination for each transmission. However, the work in [6] and the other works in [3], [4] considered that each client has the update in the packet reception status of all other clients after every cooperative transmission (referred to as full *system status update* (SSU)).

In this paper, we consider that one of the clients (referred to as the *leader*) decides the transmitting client and the packet combination that reduce the decoding delay of IDNC for each cooperative transmission. We further consider intermittent SSU at the leader such that all other clients feed back their packet reception status to the leader after several cooperative transmissions. Therefore, for all the unacknowledged transmissions, the leader decides the transmitting clients and the packet combinations without having accurate packet reception status of the other clients. To have a fair computational load on the cooperating clients, we consider that the clients take turns in becoming the leader for each round of unacknowledged transmissions. In this context, the contributions of this paper are the followings. We first extend the intermittent IDNC graph for PMP scenario in [10] to the cooperative scenario with intermittent SSU to represent all potential packet combinations for a transmitting client. We then formulate the minimum decoding delay in cooperative systems with intermittent SSU as a maximum weight clique problem over all the IDNC graphs. Since solving the formulated problem is computationally complex, we propose a heuristic algorithm that can reduce the decoding delay in the considered system. Finally, we show through simulations that the proposed algorithm can achieve a tolerable degradation compared to the full SSU scheme studied in [6].

978-1-4799-5999-0/14/$31.00 ©2014 IEEE

II. System Model and Parameters

We consider a set of clients $\mathcal{M} = [1, ..., M]$, where each client $i \in \mathcal{M}$ initially holds a subset of source packets from $\mathcal{N} = [1, ..., N]$ (denoted by $\mathcal{H}_i \subseteq \mathcal{N}$). We further consider that the clients collectively hold all the packets in $\mathcal{N}$ (i.e. $\bigcup_{i\in\mathcal{M}} \mathcal{H}_i = \mathcal{N}$). All M clients are interested in receiving all the packets of $\mathcal{N}$ and cooperate with each other to receive all the packets. In such cooperative systems, only one client can transmit a packet per time slot t and all other clients listen to the transmitted packet. Moreover, one of these clients (referred to as the *leader*) decides the transmitting clients and the packet combinations for several cooperative time slots and the leaders are nominated following the order of the client indices.

Each transmitted packet from client i to client k is subject to erasure with probability $p_{i,k}$, which is assumed to be fixed during the N packets transmission period. We assume channel reciprocity, which means that the transmitted packet from client k to client i is also subject to the same erasure probability, i.e. $p_{i,k} = p_{k,i}$. We further assume that the leaders have the knowledge of the erasure probabilities between all the clients.

At the beginning of the cooperation, each client sends a feedback to the nominated leader to acknowledge all its previously received packets. We refer to it as the *system status update* (SSU) at the nominated leader. After every T_f cooperative transmissions, each client sends a feedback to the nominated leader to acknowledge all its previously received packets. Therefore, the nominated leader selects T_f transmitting clients and T_f packet combinations for T_f transmissions at the SSU instant. We refer to it as intermittent SSU at the leaders and the period between two SSUs (i.e. T_f transmissions) as the SSU period. Such intermittent SSU creates uncertainties at the leaders about the packet reception status of the other clients during the unacknowledged transmissions. Based on these conditions, four sets of packets can be attributed to each client $k \in \mathcal{M}$ for any time slot t:

1) The *Has* set ($\mathcal{H}_k$) is defined as the set of packets received by client k and acknowledged in the last SSU.
2) The *Wants* set ($\mathcal{W}_k$) is defined as the set of missing packets at client k after the last SSU ($\mathcal{W}_k = \mathcal{N} \backslash \mathcal{H}_k$).
3) The *Uncertain* set ($\mathcal{U}_k$) is defined as the set of packets in the Wants set of client k, which are attempted after the last SSU ($\mathcal{U}_k \subseteq \mathcal{W}_k$).
4) The *Certain* set ($\mathcal{C}_k$) is defined as the set of missing packets in the Wants set of client k, which are not attempted after the last SSU ($\mathcal{C}_k = \mathcal{W}_k \setminus \mathcal{U}_k$).

In this paper, an attempted packet for client k in a transmission means this packet will be immediately decodable at client k upon receiving the transmitted packet. We refer to the packets that belong to the Has, Wants, Uncertain and Certain sets as the Has, Wants, Uncertain and Certain packets, respectively.

The leader stores the above information of all clients in an $M \times N$ *system status matrix (SSM)* $\mathbf{F} = [f_{k,l}]$, $\forall k \in \mathcal{M}, l \in \mathcal{N}$, such that: $f_{k,l} = 0$ if $l \in \mathcal{H}_k$, $f_{k,l} = 1$ if $l \in \mathcal{C}_k$ and $f_{k,l} = -1$ if $l \in \mathcal{U}_k$.

Based on the above definitions, each received packet at the clients can be one of the followings:

- *Non-innovative:* A packet is non-innovative for client k if it does not bring new information. In other words, for a non-innovative packet, all the encoded packets were previously received and decoded at client k.
- *Instantly Decodable:* A packet is instantly decodable for client k if it contains only one source packet from $\mathcal{W}_k$ that was not previously received.
- *Non-instantly Decodable:* A packet is non-instantly decodable for client k if it contains two or more source packets from $\mathcal{W}_k$ that were not previously received.

We define decoding delay similar to [6], [7], [11] as follows[1]:

Definition 1. *At any cooperative transmission, a client that still needs at least one source packet, experiences one unit increase of decoding delay if it successfully receives a packet that is either non-innovative or non-instantly decodable.*

Given the SSM for any time slot t, the leader exploits it to select a transmitting client and a packet combination that is instantly decodable to a selected subset of clients, also referred to as the *targeted clients* $\mathcal{T}$. Other clients that cannot immediately decode a missing source packet from the received packet discard it. In the considered system, the cooperative transmission is continued until all clients declare the reception of all packets in the SSU process.

Remark 1. *For T_f unacknowledged transmissions, the nominated leader selects T_f transmitting clients and T_f packet combinations at the SSU instant. Then it sends this information to the transmitting clients. Each selected transmitting client broadcasts an XORed combination of the selected source packets at its respective time slot.*

Remark 2. *In the rest of the paper, we consider an arbitrary time slot t within the SSU period and select the transmitting client and the packet combination for that time slot. The leader repeats this process T_f times for T_f unacknowledged transmissions at the SSU instant.*

III. Intermittent Local IDNC Graph

To describe all potential packet combinations that are instantly decodable by a subset of clients in cooperative systems with full SSU, the authors in [4], [6] introduced local IDNC graph. However, the limitations of such IDNC graph was illustrated in [10] for PMP scenarios with intermittent feedback. Therefore, we extend the intermittent IDNC graph, proposed in [10], to the cooperative scenario with intermittent SSU. We refer to it as *intermittent local IDNC* (IL-IDNC) graph. The leader can form an IL-IDNC graph $\mathcal{G}^i(\mathcal{V}^i, \mathcal{E}^i)$ for each client $i \in \mathcal{M}$.[2] Here, the difference with [10] is that the IL-IDNC graph $\mathcal{G}^i$ for client i includes the vertices induced by the Wants packets of other clients, which also belong to the Has set of client i, $\mathcal{H}_i$. It means that a vertex v_{kl} is added to the IL-IDNC graph $\mathcal{G}^i$, if packet $l \in \{\mathcal{H}_i \bigcap \mathcal{W}_k\}, \forall k \in \{\mathcal{M} \setminus i\}$. Once the vertices are induced, two vertices v_{kl} and v_{mn} in $\mathcal{G}^i$ can be connected by an edge in $\mathcal{E}^i$, if one of the two connectivity conditions that will be introduced in Section III-C holds.

A. Probability that a Packet is Innovative

We express the probability that any arbitrary packet $l \in \mathcal{N}$ brings new information (i.e. *innovative or was not perviously*

[1]This definition of decoding delay does not count channel inflicted delays due to erasures, but rather counts algorithmic delays.

[2]When $\mathcal{G}^i$ is formed for client i, it is considered as a transmitting client and $\mathcal{G}^i$ represents all IDNC packet combinations that can be formed by client i.

received) to any client $k \in \mathcal{M}$ as $\breve{p}_k(l)$, where

$$\breve{p}_k(l) = \begin{cases} 0 & l \in \mathcal{H}_k \\ 1 & l \in \mathcal{C}_k \\ p_k(l) & l \in \mathcal{U}_k. \end{cases} \quad (1)$$

Here, a Has packet is considered as innovative with probability equal to 0, a Certain missing packet is considered as innovative with probability equal to 1 and an Uncertain packet is considered as innovative with probability of $p_k(l)$, which can be explained as follows. Let θ_{kl} be the number of times the other transmitting clients attempt (i.e. target) client k with packet l after the last SSU. These transmitting clients are represented in the form of a vector $\mathbf{I}_{kl}$ such that each element of $\mathbf{I}_{kl}$ is a transmitting client for each of θ_{kl} attempts (i.e. $\mathbf{I}_{kl} = [i_1, ..., i_{\theta_{kl}}]$ and $|\mathbf{I}_{kl}| = \theta_{kl}$). At each one of the θ_{kl} attempts, the transmitted packet can be lost with probability $p_{i,k}, i \in \mathbf{I}_{kl}$. Consequently, the probability that Uncertain packet l is innovative for client k in time slot t is:

$$p_k(l) = \prod_{i=\mathbf{I}_{kl}(1)}^{\mathbf{I}_{kl}(\theta_{kl})} p_{i,k}. \quad (2)$$

In fact, this is the probability that Uncertain packet l is lost at client k in all of the θ_{kl} attempts.

B. Decoding Delay Increment

We start with expressing the probability that any arbitrary client k still needs at least one packet, following [10], as:

$$\bar{p}_{k,f} = 1 - \prod_{l \in \mathcal{N}} (1 - \breve{p}_k(l)), \quad \forall k \in \mathcal{M}. \quad (3)$$

Having this expression, let us now consider two vertices v_{kl} and v_{mn} in $\mathcal{G}^i$ such that $k \neq m$ and $l \neq n$. We illustrate the expected decoding delay resulting from sending packet l and packet n separately and their combination $l \oplus n$. Let $d_{kl,mn}(l)$ denote the overall decoding delay increase for clients k and m after sending packet l. When only packet l is sent, client k will experience a delay if the following three criteria hold: (1) it receives the transmitted packet l, (2) packet l was previously received (i.e. non-innovative), and (3) it still needs at least one packet. Having considered the same criteria for client m, when only packet l is sent, the expected overall decoding delay increase experienced by both clients k and m can be expressed as [10]:

$$\mathbb{E}[d_{kl,mn}(l)] = (1 - p_{i,k})(1 - \breve{p}_k(l))\bar{p}_{k,f} + (1 - p_{i,m})(1 - \breve{p}_m(l))\bar{p}_{m,f}. \quad (4)$$

Similarly, when only packet n is sent, the expression for the expected overall decoding delay increase experienced by both clients k and m can be obtained. On the other hand, when encoded packet $l \oplus n$ is sent, client k will experience a delay if the following three criteria hold: (1) it receives the transmitted packet $l \oplus n$, (2) packets l and n were previously received, i.e. non-innovative, or are still missing, i.e. non-instantly decodable (the probability of occurring this event at client k is $(1 - \breve{p}_k(l))(1 - \breve{p}_k(n)) + \breve{p}_k(l)\breve{p}_k(n)$), and (3) it still needs at least one packet. When encoded packet $l \oplus n$ is sent, the expected overall decoding delay increase experienced by both clients k and m can be expressed as:

$$\mathbb{E}[d_{kl,mn}(l \oplus n)] = (1 - p_{i,k})\{(1 - \breve{p}_k(l))(1 - \breve{p}_k(n)) + \breve{p}_k(l)\breve{p}_k(n)\}\bar{p}_{k,f} + (1 - p_{i,m})\{(1 - \breve{p}_m(l))(1 - \breve{p}_m(n)) + \breve{p}_m(l)\breve{p}_m(n)\}\bar{p}_{m,f}. \quad (5)$$

C. IL-IDNC Graph Construction

Once the vertices are generated at the IL-IDNC graph $\mathcal{G}^i(\mathcal{V}^i, \mathcal{E}^i)$ for client i, as described earlier, two vertices v_{kl} and v_{mn} are connected (i.e. adjacent) by an edge in $\mathcal{E}^i$ if one of the two connectivity conditions holds:

- **C1**: $l = n$. Packet l is needed by both clients k and m.
- **C2**: $\mathbb{E}[d_{kl,mn}(l \oplus n)] \leq \min\{\mathbb{E}[d_{kl,mn}(l)], \mathbb{E}[d_{kl,mn}(n)]\}$. Sending coded packet $l \oplus n$ is expected to achieve a lower expected decoding delay increment at clients k and m than sending packet l or packet n separately.

Given these connectivity conditions, the set of all potential packet combinations is defined by the set of all maximal cliques in IL-IDNC graph $\mathcal{G}^i$, when client i is considered as the transmitting client. Consequently, the transmitting client i can form a coded packet by XORing the source packets identified by the vertices in a maximal clique κ^i.

IV. Minimum Decoding Delay Formulation

In this section, we follow a similar approach of the minimum decoding delay formulation for PMP scenarios with intermittent feedback as in [11], and extend it to cooperative scenarios with intermittent SSU. To determine the transmitting client and the packet combination for time slot t, the leader forms M IL-IDNC graphs for M clients and selects M maximal cliques from M IL-IDNC graphs (one maximal clique per IL-IDNC graph) such that each selected maximal clique from each graph guarantees the minimum decoding delay increase with respect to that graph. Then, it selects the maximal clique among M maximal cliques that is expected to achieve the minimum decoding delay. Moreover, the leader chooses the client whose IL-IDNC graph includes the selected maximal clique as the transmitting client for time slot t. The leader uses this information to select the subsequent clients and packet combinations for the subsequent time slots within the SSU period. In order to formulate the minimum decoding delay problem for time slot t, let us assume that the leader selects client i as the transmitting client. Under this assumption, the packet selection problem can be formulated as follows.

Let κ^i be the selected maximal clique from the IL-IDNC graph $\mathcal{G}^i$ for a given time slot t. We define $\mathcal{T}(\kappa^i)$ as the set of targeted clients who have vertices in κ^i. We let $d_k(\kappa^i)$ be the decoding delay increase experienced by client k after time slot t and $D(\kappa^i)$ be the sum of the decoding delay increase of all clients, i.e. $D(\kappa^i) = \sum_{k \in \mathcal{M}} d_k(\kappa^i)$. With these considerations, the minimum expected decoding delay problem can be formulated as a maximal clique selection problem, as:

$$\kappa^i = \arg \min_{\kappa^i \in \mathcal{G}^i} \{\mathbb{E}[D(\kappa^i)]\} = \arg \min_{\kappa^i \in \mathcal{G}^i} \{\mathbb{E}[d_i(\kappa^i)] + \mathbb{E}[\sum_{k \in \{\mathcal{M} \setminus i\}} d_k(\kappa^i)]\}. \quad (6)$$

Here, the first term represents the expected decoding delay increase at the transmitting client i, which depends on the size of its Wants set since the transmitting client cannot benefit from

its own transmission. In fact, this term is independent of the selected maximal clique κ^i and thus, we can express (6) as:

$$\kappa^i = \arg \min_{\kappa^i \in \mathcal{G}^i} \{\mathbb{E}[\sum_{k \in \{\mathcal{M} \setminus i\}} d_k(\kappa^i)]\}. \quad (7)$$

Following [10], [11], we can formulate the problem in (7) as the maximum weight clique problem over the IL-IDNC graph $\mathcal{G}^i$, such that:

$$\kappa^i = \arg \max_{\kappa^i \in \mathcal{G}^i} \{\sum_{k \in \mathcal{T}(\kappa^i)} (1 - p_{i,k}) \times \breve{p}_k(l)\}. \quad (8)$$

Here, packet l is instantly decodable at targeted client k and $\breve{p}_k(l)$ is the probability that packet l is innovative for client k, which is given by (1). Having formulated the minimum decoding delay problem considering client i as the transmitting client, we now formulate the joint transmitting client and packet selection problem as follows:

$$\begin{aligned}
\kappa^{i^*} &= \arg \min_{i \in \mathcal{M}} \{\arg \min_{\kappa^i \in \mathcal{G}^i} \{\mathbb{E}[D(\kappa^i)]\}\} \\
&= \arg \min_{i \in \mathcal{M}} \{\arg \min_{\kappa^i \in \mathcal{G}^i} \{\mathbb{E}[d_i(\kappa^i)] + \mathbb{E}[\sum_{k \in \{\mathcal{M} \setminus i\}} d_k(\kappa^i)]\}\} \\
&= \arg \max_{i \in \mathcal{M}} \{\arg \max_{\kappa^i \in \mathcal{G}^i} \{\mathbb{P}[d_i(\kappa^i) = 0] + \\
&\qquad \sum_{k \in \mathcal{T}(\kappa^i)} (1 - p_{i,k}) \times \breve{p}_k(l)\}\}, \quad (9)
\end{aligned}$$

where i^* and κ^{i^*} are the selected transmitting client and the selected maximal clique, respectively, that result in the minimum expected decoding delay for time slot t. In (9), the first term represents the maximum probability that transmitting client i^* is not experiencing a decoding delay, which depends on the size of its Wants set. Furthermore, from the second term, it can be inferred that the minimum decoding delay problem for the considered cooperative system is equivalent to finding the maximum weight clique over the clients' IL-IDNC graphs, where the weight of each vertex is defined according to the packet reception probability of its inducing client and the probability that the corresponding packet is innovative for its inducing client. However, it is well known that finding the maximum weight clique of a graph is NP-hard [12]. Thus, in the next section, we will propose a heuristic algorithm to solve the problem with low complexity.

V. Joint Client and Packet Selection Algorithm

Having formulated the minimum decoding delay problem in Section IV, in this section, we design a simple heuristic algorithm that can reduce the decoding delay for time slot t. In this heuristic, the transmitting client i is first selected as follows. Generally, a client having a large Has set can form a large number of potential packet combinations for a transmission. To more efficiently consider the number of potential packet combinations, we define the importance of a Has packet of a client as the number of other clients that still need this packet. Consequently, the effective value of the Has set of each client $i \in \mathcal{M}$ can be defined as:

$$\Gamma^i = \sum_{k \in \{\mathcal{M} \setminus i\}} \left\{ \sum_{l \in \{\mathcal{H}_i \cap \mathcal{C}_k\}} 1 + \sum_{l \in \{\mathcal{H}_i \cap \mathcal{U}_k\}} p_k(l) \right\}. \quad (10)$$

In this expression, the first term counts the Certain missing packets of the other clients, which also belong to the Has set of client i, and the second term counts the Uncertain packets of the other clients (more specifically, each Uncertain packet is normalized by the probability that it is innovative), which also belong to the Has set of client i. Having the definition in (10), the leader selects the transmitting client i for time slot t, such that: $i = \arg \max_{i \in \mathcal{M}} \Gamma^i$. Once the transmitting client i is selected, the leader selects the maximal clique κ^i over the IL-IDNC graph $\mathcal{G}^i$, using the following greedy maximum weight vertex search algorithm.

Algorithm 1 Joint Client and Packet Selection (JCPS)

Calculate $\Gamma^i, \forall i \in \mathcal{M}$ using (10).
Select transmitting client $i = \arg \max_{i \in \mathcal{M}} \Gamma^i$.
Construct $\mathcal{G}^i(\mathcal{V}^i, \mathcal{E}^i)$ for transmitting client i.
Initialize $\kappa^i = \varnothing$.
Set $\mathcal{G}^i(\kappa^i) \leftarrow \mathcal{G}^i$.
While $\mathcal{G}^i(\kappa^i) \neq \varnothing$ do
Compute $w_{kl}^{\mathcal{G}^i(\kappa^i)}, \forall v_{kl} \in \mathcal{G}^i(\kappa^i)$ using (13).
Select $v_{kl}^* = \arg \max_{v_{kl} \in \mathcal{G}^i(\kappa^i)} \{w_{kl}^{\mathcal{G}^i(\kappa^i)}\}$.
Set $\kappa^i \leftarrow \kappa^i \cup v_{kl}^*$ and update subgraph $\mathcal{G}^i(\kappa^i)$.
end while

A. Packet Selection Algorithm

To select maximal clique κ^i over graph $\mathcal{G}^i(\mathcal{V}^i, \mathcal{E}^i)$, we follow the approach of greedy maximum weight vertex search algorithm proposed in [7], [11]. We first define the importance of vertex $v_{kl} \in \mathcal{V}^i$ (denoted by $\tilde{\psi}_{kl}$) as the packet reception probability of its inducing client k and the probability that the corresponding packet l is innovative for its inducing client k. This can be expressed as:

$$\tilde{\psi}_{kl} = (1 - p_{i,k}) \times \breve{p}_k(l). \quad (11)$$

To efficiently perform the greedy vertex search, the vertex's weight must not only reflect the $\tilde{\psi}_{kl}$ value, but also its adjacency to the vertices having high $\tilde{\psi}_{mn}$ values. Consequently, we define the weighted degree Δ_{kl} of vertex v_{kl} as: $\Delta_{kl} = \sum_{v_{mn} \in \mathcal{G}^i} e_{kl,mn} \tilde{\psi}_{mn}$, where $e_{kl,mn}$ is the adjacency indicator of vertices v_{kl} and v_{mn}, such that:

$$e_{kl,mn} = \begin{cases} 1 & v_{kl} \text{ is adjacent to } v_{mn} \text{ in } \mathcal{G}^i, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

We finally define the weight w_{kl} of vertex v_{kl} as:

$$w_{kl} = \tilde{\psi}_{kl} \Delta_{kl} = \{(1 - p_{i,k}) \times \breve{p}_k(l)\} \Delta_{kl}. \quad (13)$$

Having defined the weights, the maximum weight vertex search algorithm evolves as follows. Initially, there are no vertices in the selected maximal clique κ^i, i.e. $\kappa^i = \varnothing$. At the first step, the algorithm selects the vertex v_{kl}^* that has the maximum weight $w_{kl}^{\mathcal{G}^i}$ and adds it to κ^i (i.e. $\kappa^i = \{v_{kl}^*\}$). After this, the algorithm extracts the subgraph $\mathcal{G}^i(\kappa^i)$ of vertices in $\mathcal{G}^i$ that are adjacent to all previously selected vertices in κ^i. Then, it recomputes the weights of the vertices in subgraph $\mathcal{G}^i(\kappa^i)$. At the next step, it selects vertex v_{mn}^* that has the maximum weight $w_{mn}^{\mathcal{G}^i(\kappa^i)}$ and adds it to κ^i (i.e. $\kappa^i = \{\kappa^i, v_{mn}^*\}$). This process is repeated until no further vertex is adjacent to all the vertices in κ^i. Once the maximal clique κ^i is selected, the packet combination can be formed by XORing the source packets identified by the vertices

in κ^i.[3] We refer to the proposed client and maximal clique selection algorithm as *joint client and packet selection* (JCPS) algorithm. The steps of the JCPS algorithm is summarized in Algorithm 1.

To select T_f clients and T_f packet combinations for T_f unacknowledged transmissions, the leader repeats the proposed JCPS algorithm T_f times. It then sends this information to the transmitting clients. The selected transmitting clients broadcast the selected packet combinations at their respective time slots.

Remark 3. *The complexity of selecting a maximal clique κ^i based on the maximum weight vertex search is $O(M^2N)$.*

VI. Simulation Results

In this section, we present the simulation results comparing the performance of our proposed JCPS algorithm in Section V with the full SSU (FSSU) scheme, studied in [6], that considers full SSU after each cooperative transmission. The FSSU scheme first forms local IDNC graphs for all clients and attributes a weight to each vertex in the local graphs. Then it determines the maximum weight vertex among the vertices in all the clients' local graphs and selects its respective client as the transmitting client. Finally, it uses the maximum weight vertex search to select the maximal clique.

In the simulations, the mean decoding delay shows the average over the complete reception of N packets by M clients. Initially, each client holds a subset of N packets in the range from 55% to 95% with an average equal to 75%. Here, the packet erasure probabilities of the channels among the clients range from $[0.05, 0.25]$ with an average equal to 0.15. Fig. 1 illustrates the mean decoding delay achieved by different algorithms against SSU period T_f (for $N = 30$, $M = 30$). As expected, the degradation in mean decoding delay obtained in the JCPS algorithm increases with the increase of the SSU period compared to the FSSU scheme. When the FSSU scheme requires the SSU after each transmission, the number of SSUs for the proposed JCPS algorithm decreases with increasing the SSU period. Moreover, the performance degradation of the proposed algorithm is tolerable, given it requires less number of SSUs compared to the FSSU scheme. Figs. 2(a) and 2(b) further illustrate the mean decoding delay achieved by different algorithms against the number of clients M (for $N = 30$ and $T_f = 5$) and number of packets N (for $M = 30$ and $T_f = 5$), respectively.

VII. Conclusion

We studied the problem of decoding delay reduction for IDNC in broadcast cooperative systems with intermittent SSU. We first described an IL-IDNC graph to define all potential packet combinations for a transmitting client. We then formulated the joint client and packet selection problem that result in the minimum expected decoding delay for each cooperative transmission as a maximum weight clique problem over all the IL-IDNC graphs. Since solving the formulated problem was computationally complex, we proposed a heuristic algorithm to select the transmitting client and the packet combination that can reduce the decoding delay in each transmission. Simulation results showed that the decoding delay performance of our proposed algorithm is close to that of the scheme with full SSU while using a smaller number of SSUs.

[3]The leader uses this information to select the transmitting clients and the packet combinations for the subsequent time slots within the SSU period.

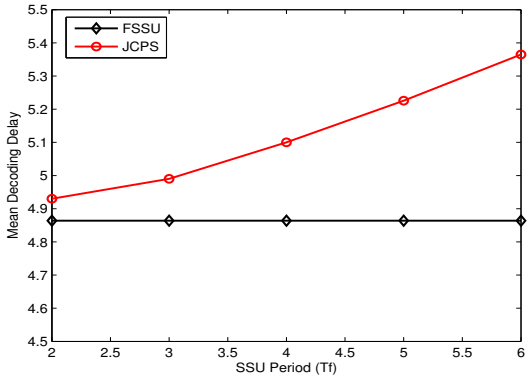

Fig. 1. Mean decoding delay versus SSU period T_f for M = N = 30.

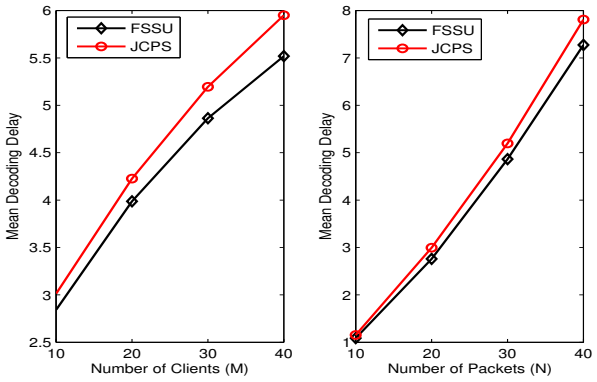

Fig. 2. Mean decoding delay versus (a) number of clients M for $N = 30$ and $T_f = 5$ (b) number of packets N for $M = 30$ and $T_f = 5$.

References

[1] Q. Zhang, J. Heide, M. V. Pedersen, and F. H. Fitzek, "MBMS with user cooperation and network coding," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–6.

[2] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou, "Cooperative video streaming on smartphones," in *49th Annual Allerton Conference on Communication, Control, and Computing*, 2011, pp. 220–227.

[3] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *IEEE Information Theory Workshop*, 2010, pp. 1–5.

[4] S. E. Tajbakhsh, P. Sadeghi, and R. Kennedy, "Centralized and cooperative transmission of secure multiple unicasts using network coding," 2013. [Online]. Available: http://arxiv.org/abs/1403.6143

[5] A. Douik, S. Sorour, H. Tembine, T. Y. Al-Naffouri, and M.-S. Alouini, "A game-theoretic framework for decentralized cooperative data exchange using network coding," 2014. [Online]. Available: http://arxiv.org/abs/1404.3637

[6] N. Aboutorab, P. Sadeghi, and S. E. Tajbakhsh, "Instantly decodable network coding for delay reduction in cooperative data exchange systems," in *IEEE International Symposium on Information Theory (ISIT)*, 2013, pp. 3095–3099.

[7] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *IEEE Global Telecommunications Conference(GLOBECOM)*, 2010, pp. 1–5.

[8] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Fourth Workshop on Network Coding, Theory and Applications (NetCod)*, 2008, pp. 1–6.

[9] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *International Symposium on Network Coding (NetCod)*, 2013, pp. 1–6.

[10] A. Douik, S. Sorour, T. Al-Naffouri, and M. Alouini, "A lossy graph model for delay reduction in generalized instantly decodable network coding," *IEEE Wireless Commun. Lett.*, 2014.

[11] S. Sorour, A. Douik, S. Valaee, T. Y. Al-Naffouri, and M.-S. Alouini, "Partially blind instantly decodable network codes for lossy feedback environment," 2013. [Online]. Available: http://arxiv.org/abs/1307.3791

[12] M. R. Garey, *Computers and intractability*. freeman New York, 1979.