Instantly Decodable Network Coding for Delay Reduction in Cooperative Data Exchange Systems

Neda Aboutorab Parastoo Sadeghi Shahriar Etemadi Tajbakhsh
Research School of Engineering, The Australian National University, Canberra, Australia
Email: {neda.aboutorab, parastoo.sadeghi, shahriar.etemadi-tajbakhsh}@anu.edu.au

***Abstract*—This paper investigates the use of instantly decodable network coding (IDNC) for minimizing the mean decoding delay in multicast cooperative data exchange systems, where the clients cooperate with each other to obtain their missing packets. Here, IDNC is used to reduce the decoding delay of each transmission across all clients. We first introduce a new framework to find the optimum client and coded packet that result in the minimum mean decoding delay. However, since finding the optimum solution of the proposed framework is NP-hard, we further propose a heuristic algorithm that aims to minimize the lower bound on the expected decoding delay in each transmission. The effectiveness of the proposed algorithm is assessed through simulations.**

I. Introduction

Imagine that a group of *geographically close* wireless clients are interested in downloading the same set of packets from the base station. We assume that these clients have collectively received the set of the packets broadcast from the base station but individually missed some packets. In such scenarios, instead of solely relying on the base station to complete their download, an increasingly attractive strategy is for the clients to cooperate with each other by exchanging network coded packets until all of them have received their requested packets [1]–[6]. The main advantage of such cooperative scheme is reducing the load of the base station. In addition, short-range transmission links among the clients can potentially result in a faster, more reliable and cheaper delivery of the information to them.

In this paper, we consider the problem of cooperative multicast exchange of packets to a set of clients. We assume that a set of clients are interested in receiving a subset of packets broadcast from the base station. Here, we consider the applications in which the received packets should bring new information to the clients, regardless of their reception order. Examples of such applications are multiple description coded systems and sensor/emergency networks [7]. In such applications, if a received packet at a client does not bring new information or cannot be instantly decoded, that client will experience *decoding delay*.

To minimize the decoding delay of the clients, a possible strategy is to employ instantly decodable network coding (IDNC) [7]–[11] schemes and encode the transmitted packets such that the decoding delay across all clients is minimized in each transmission. IDNC is an important subclass of network coding in which the sender exploits the diversity of the received and lost packets at different clients for making network coding decisions that provide instant packet decodability at a subset or all clients upon successful packet reception. In addition to being instantly decodable, the low-complexity XOR-based encoding and decoding process of IDNC, has encouraged its potential application in multicast or broadcast transmissions.

IDNC can be further divided into two categories, strict IDNC (SIDNC) [7]–[9] and the generalized IDNC (GIDNC) [10], [11]. In SIDNC, the coded packets are forced to include *at most* one missing source packet for each client. So, a client will either receive one or no new packet in each IDNC transmission. This limits the coding opportunities and can increase the decoding delay. In contrast, in GIDNC, some receivers may receive more than one new packet, which will be discarded. Thus, GIDNC, by appropriately selecting a subset (or if possible, all) of the clients to be addressed by one of their missing packets in each transmission, potentially reduces the decoding delay.

The minimum mean decoding delay problem in the non-cooperative IDNC broadcast systems is investigated in [7], [9], [10], [12]. However, in this paper, our goal is to minimize the mean decoding delay in GIDNC multicast cooperative data exchange systems. Unlike [7], [9], [10], [12], where the base station knows all the packets and can broadcast any desirable combination of them to the clients, in the cooperative scenario each client possesses only a subset of packets and can broadcast the combinations of those packets to the other clients. Since different clients may possess different packets, minimizing the mean decoding delay in such systems raises the following question: *Which client and what combination of the packets should be selected for the transmission so that the mean decoding delay across all clients is minimized?*

To answer this question, we first formulate the optimum client and packet selection scheme that achieves the minimum mean decoding delay in GIDNC cooperative systems. Then, it will be shown that finding the optimum client and packet selection in such cooperative scenarios is equivalent to the maximum weighted clique search over all the cliques in the clients' local weighted GIDNC graphs. However, since finding the optimum clique is NP-hard, we use the obtained properties of the optimal solution and propose a heuristic client and packet selection algorithm. We then assess the performance of the proposed algorithm by comparing it to the existing algorithms in the literature. The simulation results show that the proposed algorithm always outperforms the other investigated schemes.

The distinctive aspects of this paper are as follows. To the best of our knowledge, minimizing the mean decoding delay for the cooperative scenarios has not been considered in the literature. In addition, most of the existing works consider the broadcast scenario [1]–[5], where all the clients are interested in

This work was supported under Australian Research Councils Discovery Projects funding scheme (project no. DP120100160).

all the packets. However, our work is more general, as broadcast can be considered as a special case of multicast. In addition, unlike [1]–[5] that assume erasure-free links among the clients, in this paper, these links are considered to be subject to packet erasures, where the probabilities of erasure can be different from link to link. Furthermore, this work is among the first works in the literature [6], [13], where an XOR-based coding and decoding scheme, e.g. GIDNC, is considered as the transmission framework for the cooperative exchange of packets.

II. System Model and Parameters

Here, we consider the problem of data exchange between a group of wireless clients. We assume that a set of packets $\mathcal{N} = [1, ..., N]$ is supposed to be delivered to a set of clients $\mathcal{M} = [1, ..., M]$. Each client is interested in a subset of packets in $\mathcal{N}$. We will refer to the requested and unrequested packets of each client by its *primary* and *secondary* packets, respectively. In the initial phase of transmission, the base station broadcasts the N packets of the frame uncoded. We assume that each transmitted packet from the base station is subject to erasure with the probability of p_i at client i. Once the initial phase of the transmission is completed, the cooperative phase is started. We assume each client i knows a subset of packets, denoted by $\mathcal{H}_i \subseteq \mathcal{N}$, and the clients collectively know all the packets in $\mathcal{N}$, i.e. $\cup_{i\in\mathcal{M}}\mathcal{H}_i = \mathcal{N}$.

In the cooperative transmission phase, each sent packet from client i to client j is subject to erasure with the probability $p_{i,j}$, which is assumed to be fixed during a frame transmission period. Furthermore, we assume that the transmission in the reverse link, i.e. the transmitted packet **from** client j to client i is also subject to the same erasure probability, i.e. $p_{i,j} = p_{j,i}$. We define the demand ratio μ_i of client i as the ratio of the number of packets requested by this client to the total number of packets in the frame, N. It can be easily inferred that broadcast can be considered to be a special case of multicast, where the demand ratio of all clients are equal to 1. The average demand ratio of all the clients can then be expressed as $\mu = \frac{\sum_{i=1}^{M}\mu_i}{M}$.

We assume that each client listens to all the packet transmissions and stores the successfully received primary and secondary packets. We also assume that each client knows the index of the primary and secondary packets that are available to the other clients. Based on the above conditions, three sets of packets can be associated to each client i.

- The *Has* set (denoted by $\mathcal{H}_i$) is the set of the packets successfully received by client i. This set includes both the primary and secondary packets received by this client.
- The *Lacks* set (denoted by $\mathcal{L}_i$) is the set of the packets that are not successfully received by client i, whether primary or secondary. In other words, $\mathcal{L}_i = \mathcal{N} \setminus \mathcal{H}_i$.
- The *Wants* set (denoted by $\mathcal{W}_i$) is the set of the primary packets that are not successfully received by client i. Note that $\mathcal{W}_i \subseteq \mathcal{L}_i$.

Each client then stores this information in a state feedback matrix (SFM) $\mathbf{F} = [f_{kl}], \forall k \in \mathcal{M}, l \in \mathcal{N}$ as, $f_{kl} = 0$ for all $l \in \mathcal{H}_k$, $f_{kl} = 1$ for all $l \in \mathcal{W}_k$, and $f_{kl} = -1$, otherwise.

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

1) Non-innovative packet: A packet is non-innovative for client i if it contains no source packets from $\mathcal{W}_i$.
2) Instantly decodable packet: A packet is instantly decodable for client i if it contains only one source packet from $\mathcal{W}_i$.
3) Non-instantly decodable packet: A packet is non-instantly decodable for client i if it contains two or more source packets from $\mathcal{L}_i$.

We define the decoding delay similar to [7] and [10] as follows:

Definition 1. At any cooperative phase transmission, a client i, with a non-empty Wants set, experiences one unit increase of *decoding delay* if it successfully receives a packet that is a non-innovative or non-instantly decodable packet.

Once the above SFM is formed at the clients, they use it to cooperate with each other by transmitting GIDNC packets to an appropriately selected subset (or if possible, all) of the clients, also referred to as the *targeted clients*, $\mathcal{T}$. We further refer to the clients targeted by one of their primary missing packets as the *primary targeted clients*, $\mathcal{T}_p$, and the clients targeted by one of their missing secondary packets by *secondary targeted clients*, $\mathcal{T}_s$. In order to decide which client sends what combination of its received packets to the other clients, we introduce a local GIDNC graph representation at each client, and then propose a packet selection policy that will result in the minimum mean decoding delay for the cooperative transmissions.

III. Local GIDNC Graphs at the Clients

The GIDNC graph representation for the non-cooperative multicast systems was first introduced in [11]. Later, the authors in [13] extended the GIDNC graph representation in [11] to the cooperative scheme, where each client i forms a local GIDNC graph $\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$. Here, the only difference with [11] is that the local graph of client i will only include the vertices induced by the missing packets of the 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 local graph of client i, if packet $l \in (\mathcal{H}_i \cap \mathcal{L}_k)$, $\exists k \in \mathcal{M}$. Once the vertices are formed at the local graph of client i, two vertices v_{kl} and v_{mn} are connected by an edge in $\mathcal{E}_i$ if one of the following conditions is true:

- C1: $l = n \Longrightarrow$ The two vertices are induced by the same missing packet l by two different clients k and m. The generated edge by this condition does not involve any packet combination.
- C2: $l \in \mathcal{H}_m$ and $n \in \mathcal{H}_k \Longrightarrow$ The missing packet corresponding to each vertex belongs to the Has set of the client that induced the other vertex. The generated edge by this condition represents a packet combination of $l \oplus n$.

However, this graph representation is more suitable when optimizing the packets combination for a broadcast scenario, where all the clients are interested in receiving all the packets. The optimization of packets combination in a multicast scenario necessitates the delivery of all possible secondary packets to the clients that do not belong to $\mathcal{T}_p$. Although these packets are not required to be received at these clients, their reception will enlarge these clients' Has sets. This will increase the coding opportunities for these clients in the future steps, as required by condition C2. However, the inclusion of these packets in each step should not affect the instant decodability of the primary packets at the clients that belong to $\mathcal{T}_p$. To achieve this goal, we introduce our two-layered GIDNC graph design for each client i. The primary layer $\mathcal{G}_p^i(\mathcal{V}_p^i, \mathcal{E}_p^i)$ consists of the primary packets still requested by the other clients that belong to the Has set of client i, and the secondary layer $\mathcal{G}_s^i(\mathcal{V}_s^i, \mathcal{E}_s^i)$ is constructed by generating a vertex $v_{kl} \in \mathcal{V}_s^i$ in the secondary graph of client i for each packet $l \in (\mathcal{H}_i \cap (\mathcal{L}_k \setminus \mathcal{W}_k))$, $\exists k \in \mathcal{M}$. We will refer to the primary layer as the *primary graph*, the secondary layer as the *secondary graph*, and the union of the primary and secondary layers as the *complete graph* at client i.

IV. Problem Formulation

In this section, we follow the proposed approach in [10] and [12] to formulate the minimum mean decoding delay problem for the non-cooperative broadcast scenarios, and extend it to multicast cooperative data exchange scenarios. As discussed in Section III, in the considered cooperative scenario, each client forms a local graph, where based on this graph, it can find GIDNC packets to broadcast to the other clients. *However, another important question here is that in order to minimize the mean decoding delay, which client should be selected for the transmission at each time-slot*. In order to answer this question, let us first assume that client k is selected for the transmission. Under this assumption, the minimum mean decoding delay problem can be formulated as follows.

Let $\kappa^k = \kappa_p^k \cup \kappa_s^k$ be the selected maximal clique (from the local graph of client k) for a given transmission at time-slot t at client k, where κ_p and κ_s are the maximal cliques of the primary and the secondary graphs, respectively. We define $\mathcal{T}(\kappa^k)$ as the set of targeted clients in this transmission (i.e. the set of clients having vertices in κ^k). Furthermore, the primary targeted clients in this transmission are denoted by $\mathcal{T}_p(\kappa_p^k)$. We let $d_i(\kappa^k)$ be the decoding delay increase experienced by client i for this transmission. If client i has a vertex in κ_p^k, it will never experience a decoding delay increase (i.e. $d_i(\kappa^k) = 0$). If not, then $d_i(\kappa^k)$ can be either 0 or 1 with the following probabilities: $P[d_i(\kappa^k) = 0] = p_{k,i}$, and $P[d_i(\kappa^k) = 1] = 1 - p_{k,i}$. It should be noted that at transmitting client k, the decoding delay increase can be written as, $d_k(\kappa^k) = 1$ if $\mathcal{W}_k \neq \varnothing$, and $d_k(\kappa^k) = 0$ if $\mathcal{W}_k = \varnothing$. It means that client k will experience a unit increase in its decoding delay if it has a non-empty Wants set, since it cannot benefit from its own transmission.

We let $D(\kappa^k)$ be the sum of the decoding delay increases of all clients after this transmission. Then, we will have

$$D(\kappa^k) = \sum_{i \in \mathcal{M}_w} d_i(\kappa^k) \tag{1}$$

where $\mathcal{M}_w$ is the set of clients having non-empty Wants sets. Based on these definitions, the expected sum decoding delay increase after this transmission can be expressed as:

$$E[D(\kappa^k)] = d_k(\kappa^k) + \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}_p(\kappa_p^k)} (1 - p_{k,i}) \tag{2}$$

It should be noted that the primary clique is the one that controls the decoding delay in each transmission. Consequently, we can formulate the minimum mean decoding delay problem as a clique selection algorithm, such that:

$$\begin{aligned} \kappa_p^k &= \arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{E[D(\kappa_p^k)]\} \\ &= \arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{d_k(\kappa_p^k) + \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}_p(\kappa_p^k)} (1 - p_{k,i})\} \end{aligned} \tag{3}$$

Here, since the term $d_k(\kappa_p^k)$ is only dependent on the size of the Wants set of transmitting client k and is independent of the selected clique κ_p^k, (3) can be further written as

$$\kappa_p^k = \arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{ \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}_p(\kappa_p^k)} (1 - p_{k,i})\} \tag{4}$$

Having solved the minimum mean decoding delay problem at each client, we can now express the joint client and packet selection for the cooperative system as

$$\begin{aligned} \kappa_p^{k^*} &= \arg\min_{k \in \mathcal{M}} \{\arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{E[D(\kappa_p^k)]\}\} \\ &= \arg\min_{k \in \mathcal{M}} \{\arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{d_k(\kappa_p^k) \\ &+ \sum_{i \in \mathcal{M}_w \setminus \mathcal{T}_p(\kappa_p^k), i \neq k} (1 - p_{k,i})\}\} \\ &= \arg\min_{k \in \mathcal{M}} \{\arg\min_{\kappa_p^k \in \mathcal{G}_p^k} \{d_k(\kappa_p^k) \\ &+ \sum_{i \in \mathcal{M}_w, i \neq k} (1 - p_{k,i}) - \sum_{i \in \mathcal{T}_p(\kappa_p^k)} (1 - p_{k,i})\}\} \end{aligned} \tag{5}$$

where k^* and $\kappa_p^{k^*}$ are the selected client for the transmission and the optimum primary clique form its local graph, respectively, that result in the minimum mean decoding delay in the system. From (5), it can be concluded that the minimum mean decoding delay problem for the considered cooperative system is equivalent to finding all the cliques in the clients' local graphs, and choosing the clique that results in the minimum mean decoding delay among them. However, it is well known that finding all the cliques in the local graphs of the clients is NP-hard. Thus, in the next section, we will design a simple heuristic algorithm to solve the problem with much lower complexity.

V. The Proposed Heuristic Algorithm

Having the minimum mean decoding delay problem formulated in Section IV, we follow the same lines of thought as in [12] and extend it to reduce the mean decoding delay in the cooperative data exchange system. To be able to design such an algorithm, we first define $a_{ij,mn}$ to be the adjacency indicator of vertices v_{ij} and v_{mn} in graph $\mathcal{G}^k$ such that $a_{ij,mn}^k = 1$, if v_{ij} is connected to v_{mn} in $\mathcal{G}^k$, and $a_{ij,mn}^k = 0$, otherwise.

The proposed algorithm starts by finding the lower bound on the potential decoding delay that may result from selecting each primary vertex from clients' local graphs. At the beginning, assuming that client $k \in \mathcal{M}$ is chosen for the transmission to the other clients and no vertices is selected yet, i.e. $\kappa_p^k = \varnothing$, the expected decoding delay is equal to

$$\tilde{D}_k^{(0)} = d_k + \sum_{i \in \mathcal{M}_w, i \neq k} (1 - p_{k,i}) \tag{6}$$

Note that according to Definition 1, an erasure at client i does not increase the decoding delay. In other words, Definition 1 controls the algorithmic decoding delays.

Algorithm 1 The Proposed Algorithm

```
1) Initialize Construct the local graph $\mathcal{G}^k, \forall k \in \mathcal{M}$.
   Calculate $\tilde{D}_k^{(0)}, \forall k \in \mathcal{M}$ by using (6).
   Calculate $\tilde{D}_{k,ij}^{(1),LB}$, using (7), for each primary vertex $v_{ij} \in \mathcal{G}_p^k, \forall k \in \mathcal{M}$. Select $k^*, v_{ij}^*$ as shown in (9).
   For the selected client $k^*$, let $\kappa_p^{k^*} = \{v_{ij}^*\}$ and $\kappa_s^{k^*} = \varnothing$.
2) While $\mathcal{G}_p(\kappa_p^{k^*}) \neq \varnothing$ do
   Compute $\Psi_{mn}^{\mathcal{G}_p^{k^*}}, \forall v_{mn} \in \mathcal{G}_p(\kappa_p^{k^*})$ using (10).
   Select $v_{mn}^* = \arg\max_{v_{mn} \in \mathcal{G}_p(\kappa_p^{k^*})} \{\Psi_{mn}^{\mathcal{G}_p^{k^*}}\}$.
   Set $\kappa_p^* \leftarrow \kappa_p^{k^*} \cup v_{mn}^*$, and update subgraph $\mathcal{G}_p(\kappa_p^{k^*})$.
3) While $\mathcal{G}_s(\kappa_s^{k^*}) \neq \varnothing$ do
   Compute $\Psi_{mn}^{\mathcal{G}_s^{k^*}}, \forall v_{mn} \in \mathcal{G}_s(\kappa_s^{k^*})$ using (10).
   Select $v_{mn}^* = \arg\max_{v_{mn} \in \mathcal{G}_s(\kappa_s^{k^*})} \{\Psi_{mn}^{\mathcal{G}_s^{k^*}}\}$.
   Set $\kappa_s^{k^*} \leftarrow \kappa_s^{k^*} \cup v_{mn}^*$, and update subgraph $\mathcal{G}_s(\kappa_s^{k^*})$.
4) $\kappa^{k^*} = \kappa_p^{k^*} \cup \kappa_s^{k^*}$. Transmit the packet identified by $\kappa^{k^*}$ by client $k^*$.
```

Note that this step corresponds to the case when no client is targeted yet. Assuming that a primary vertex v_{ij} is selected and added to the clique κ_p^k, the lower bound on the potential decoding delay that may result from selecting this vertex can be expressed as

$$\tilde{D}_{k,ij}^{(1),LB} = \tilde{D}_k^{(0)} - \Psi_{ij}^{\mathcal{G}_p^k} \tag{7}$$

where $\Psi_{ij}^{\mathcal{G}_p^k}$ is the weight of vertex v_{ij} defined as

$$\Psi_{ij}^{\mathcal{G}_p^k} = \sum_{l \in \{i \cup \mathcal{M}_{ij}^{\mathcal{G}_p^k}\}} (1 - p_{k,l}) \tag{8}$$

Here, $\mathcal{M}_{ij}^{\mathcal{G}_p^k}$ is the set of clients that have at least one vertex adjacent to v_{ij} in $\mathcal{G}_p^k$. It can be easily inferred that any client which does not have any primary vertex adjacent to v_{ij} will not be targeted by this transmission. Furthermore, in the best case scenario, if vertex v_{ij} forms a clique with all $|\mathcal{M}_{ij}^{\mathcal{G}_p^k}|$ vertices, then all the clients in $\mathcal{M}_{ij}^{\mathcal{G}_p^k}$ will be targeted and they will not experience any decoding delay in this time-slot. Thus, under this scenario, $\tilde{D}_{k,ij}^{(1),LB}$ will be the exact expected decoding delay that will result from this transmission. In all other cases, it will be a lower bound on the decoding delay. Once $\tilde{D}_{k,ij}^{(1),LB}$ is calculated for all vertices in all clients' local graphs, the algorithm chooses client k^* with the minimum lower bound on the potential decoding delay as

$$k^*, v_{ij}^* = \arg \min_{k \in \mathcal{M}, v_{ij} \in \mathcal{G}_p^k} \{\tilde{D}_{k,ij}^{(1),LB}\} \tag{9}$$

Client k^* is then chosen to transmit its maximum weight clique to the other clients. In order to form the optimum packet selection, this algorithm performs maximum weight vertex search. It starts by choosing v_{ij}^* to be added to the selected clique $\kappa_p^{k^*}$. Then the algorithm extracts the subgraph $\mathcal{G}_p^{k^*}(\kappa_p^{k^*})$, which consists of vertices in $\mathcal{G}_p^{k^*}$ that are adjacent to all the vertices in $\kappa_p^{k^*}$. It then recomputes the weights of vertices in $\mathcal{G}_p^{k^*}(\kappa_p^{k^*})$ as follows:

$$\Psi_{mn}^{\mathcal{G}_p^{k^*}} = \sum_{i \in \mathcal{T}_p(\kappa_p^{k^*})} (1 - p_{k^*,i}) + \sum_{l \in \{m \cup \mathcal{M}_{mn}^{\mathcal{G}_p^{k^*}}\}} (1 - p_{k^*,m}) \tag{10}$$

Then, in Step 2, the algorithm chooses vertex v_{mn}^* with the maximum weight $\Psi_{mn}^{\mathcal{G}_p^{k^*}}$. Once v_{mn}^* is chosen to be added to the clique, the new lower bound on the expected decoding delay becomes:

$$\begin{aligned} \tilde{D}_{k^*,mn}^{(2),LB} &= \tilde{D}_{k^*}^{(0)} - \Psi_{mn}^{\mathcal{G}_p^{k^*}} \\ &= \tilde{D}_{k^*,ij}^{(1),LB} + \sum_{l \in \{\mathcal{M}_{ij}^{\mathcal{G}_p^{k^*}} \backslash (m \cup \mathcal{M}_{mn}^{\mathcal{G}_p^{k^*}})\}} (1 - p_{k^*,l}) \end{aligned} \tag{11}$$

Since $m \cup \mathcal{M}_{mn}^{\mathcal{G}_p^{k^*}}$ is already a subset of $\mathcal{M}_{ij}^{\mathcal{G}_p^{k^*}}$, the larger $\Psi_{mn}^{\mathcal{G}_p^{k^*}}$, the smaller the second term in (11), and thus the smaller the increase on the expected decoding delay lower bound.

After adding the new vertex v_{mn}^* to the clique $\kappa_p^{k^*} = \{v_{ij}^*, v_{mn}^*\}$, we extract the new subgraph $\mathcal{G}_p^{k^*}(\kappa_p^{k^*})$, whose vertices are adjacent to all the vertices in $\kappa_p^{k^*}$. The algorithm then recomputes the weights of the vertices in $\mathcal{G}_p^{k^*}(\kappa_p^{k^*})$ as in (10) and the vertex with the maximum weight is selected and added to $\kappa_p^{k^*}$. This process is repeated until no further vertices are adjacent to all the vertices in $\kappa_p^{k^*}$.

Once the primary clique $\kappa_p^{k^*}$ is selected at client k^*, the same process will be repeated on $\mathcal{G}_s^{k^*}$ to find the secondary clique $\kappa_s^{k^*}$ that is adjacent to all vertices in $\kappa_p^{k^*}$. Then, the complete maximal clique κ^{k^*} can be found to be the union of $\kappa_p^{k^*}$ and $\kappa_s^{k^*}$, i.e. $\kappa^{k^*} = \kappa_p^{k^*} \cup \kappa_s^{k^*}$. The algorithm is summarized in Algorithm 1. Once the clique is computed, the selected client, k^*, forms and sends the coded packet by XORing the source packets identified by the vertices in κ^{k^*}. The whole algorithm is then repeated until all clients receive all the missing packets in their Wants sets. It can be shown that the complexity of the proposed algorithm is of the order of $O(M^2N)$.

VI. Simulation Results

In this section, we present the simulation results comparing the performance of our proposed algorithm, denoted by 'Proposed' in the figures' legends, to three other algorithms.

The first algorithm is the random client selection algorithm, denoted by 'Random', where initially a random client is selected and then Steps 2 to 4 of Algorithm 1 are performed on its local graph to form the coded packet. Furthermore, we have extended the proposed non-cooperative SIDNC packet selection algorithm in [7] to the cooperative data exchange scenario. Here, the proposed algorithm in [7] is extended such that first at each client $k, \forall k \in \mathcal{M}$, a weight is associated to each packet i, where $i \in (\mathcal{H}_k \cap \mathcal{W}_m), \exists m \in \mathcal{M}$. This weight is a measure of packet importance and reflects the number of clients that still need packet i. Then the packet with the maximum weight, and consequently its respective client, are selected for

Another possible policy is to first find the maximum weight cliques and consequently their respective lower bounds on the expected decoding delay at all clients' local graphs, and then choose client k^* that results in the minimum lower bound. However, this scheme is more complex as the maximum weight cliques need to be found at all clients.

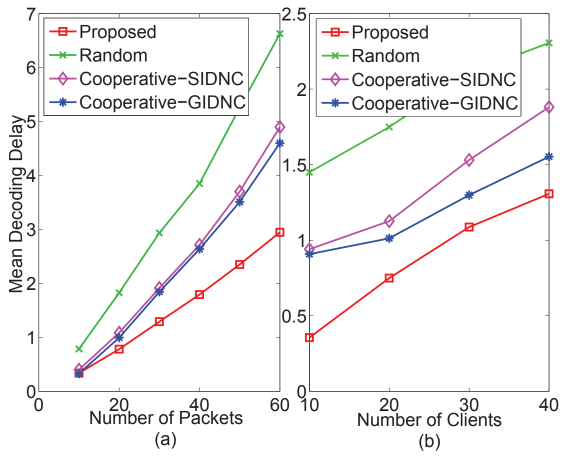

Figure 1: Decoding delay versus (a) number of packets N for $M = 20$ clients, (b) number of clients M for $N = 20$ packets

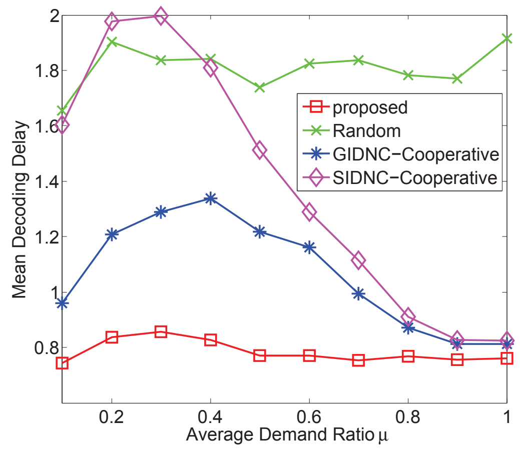

Figure 2: Decoding delay versus the average demand ratio μ for $M = N = 20$ clients and packets

the transmission. Next, from all the remaining packets that do not violate the SIDNC constraint of the selected packet, the one with the maximum weight is chosen to be XORed with the previously selected packet. This process is then repeated until no more packet can be XORed with the selected set of packets, due to the SIDNC constraint. In addition, we have also compared our proposed algorithm with the algorithm in [13], in which first a local graph is formed at each client and then a weight is attributed to each vertex in the local graphs. Then, among the vertices in all the clients' local graphs, the vertex with the maximum weight, and consequently its respective client, is selected for the transmission. Next, the maximum weight vertex search is performed on the selected client's local graph to find the maximal clique that includes the selected vertex. In the proposed algorithm in [13], the weight of a vertex shows the size of its Wants sets as well as its adjacency to the clients with large Wants sets. These schemes are referred to as 'Cooperative-SIDNC' and 'Cooperative-GIDNC', in the figures' legends, respectively.

In the simulations, we assume that the packet erasure probabilities of the links between the base station and the clients range from [0 to 0.3] with an average of 0.15, and the packet erasure probabilities of the links among the clients range from [0 to 0.1] with an average of 0.05. Furthermore, it is assumed that these probabilities are fixed during the transmission of a frame and can change from frame to frame. Figure 1(a) and (b) depict the mean decoding delays achieved by different algorithms against the number of clients M and the number of packets N, for $N = 20$ and $M = 20$, respectively. These figures show that the proposed algorithm always outperforms the other investigated algorithms. Figure 2 illustrates the mean decoding delays achieved by different algorithms against the average demand ratio μ for $M = N = 20$ clients and packets. As it can be seen, the proposed algorithm outperforms the other investigated algorithms for the entire range of μ.

VII. Conclusions

In this paper, we proposed a new framework for the instantly decodable multicast cooperative transmission, in which the clients cooperate with each other to obtain their missing packets. Here, in order to reduce the decoding delay of the clients, IDNC is employed to form the coded packets that are instantly decodable at the clients. Furthermore, we formulated the optimum client and coded packet selection that result in the minimum mean decoding delay. However, since finding the optimum client and the coded packet is NP-hard, we proposed a new heuristic algorithm that reduces the decoding delay in each transmission. Extensive simulation results confirm the effectiveness of the proposed algorithm.

References

[1] S. Raza, D. Li, C. N. Chuah, and G. Cheung, "Cooperative peer-to-peer repair for wireless multimedia broadcast," in *Proc. IEEE ICME*, Jul. 2007.

[2] S. E. Rouayheb, A. Sprinston, and P. Sadeghi, "On coding for cooperative data exchange," in *Proc. IEEE Inf. Theory Workshop (ITW'10)*, Jan. 2010.

[3] A. Sprinston, P. Sadeghi, G. Booker, and S. E. Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT'10)*, Jun. 2010.

[4] T. Courtade, B. Xie, and R. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *Proc. IEEE Military Commun. Conf. (MILCOM'10)*, Nov. 2010.

[5] S. E. Tajbakhsh, P. Sadeghi, and R. Shams, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *Proc. Workshop on Network Coding, Theory and App. (NETCOD'11)*, Jul. 2011.

[6] S. E. Tajbakhsh and P. Sadeghi, "Coded cooperative data exchange for multiple unicasts," in *Proc. IEEE Inf. Theory Workshop (ITW'12)*, 2012.

[7] 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 Commun. and Netw.*, pp. 1–14, Jan. 2010.

[8] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. Workshop on Network Coding, Theory and App. (NETCOD'8)*, 2008.

[9] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proc. Workshop on Network Coding, Theory and App. (NETCOD'09)*, 2009.

[10] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'10)*, Dec. 2010.

[11] ——, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. IEEE Int. Conf. on Commun. (ICC'10)*, 2010.

[12] S. Sorour, N. Aboutorab, P. Sadeghi, M. S. Karim, T. Al-Naffouri, and M. Alouini, "Delay reduction in persistent erasure channels for generalized instantly decodable network coding," *to appear in Proc. IEEE Vehicular Technology Conferecne (VTC'13)*, Jun. 2013.

[13] S. E. Tajbakhsh, P. Sadeghi, and R. Kennedy. Centralized and cooperative transmission of secure multiple unicasts using network coding. [Online]. Available: http://arxiv.org/abs/1305.1415