Decoding Delay Controlled Reduction of Completion Time in Instantly Decodable Network Coding

Ahmed Douik, *Student Member, IEEE*, Sameh Sorour, *Member, IEEE*,
Tareq Y. Al-Naffouri, *Member, IEEE*, and Mohamed-Slim Alouini, *Fellow, IEEE*

Abstract—For several years, the completion time and the decoding delay problems in Instantly Decodable Network Coding (IDNC) were considered separately and were thought to completely act against each other. Recently, some works aimed to balance the effects of these two important IDNC metrics but none of them studied a further optimization of one by controlling the other. In this paper, we study the effect of controlling the decoding delay to reduce the completion time below its currently best known solution in persistent erasure channels. We first derive the decoding-delay-dependent expressions of the users' and overall completion times. Although using such expressions to find the optimal overall completion time is NP-hard, we design two novel heuristics that minimizes the probability of increasing the maximum of these decoding-delay-dependent completion time expressions after each transmission through a layered control of their decoding delays. We, then, extend our study to the limited feedback scenario. Simulation results show that our new algorithms achieves both a lower mean completion time and mean decoding delay compared to the best known heuristic for completion time reduction. The gap in performance becomes significant for harsh erasure scenarios.

Index Terms—Instantly decodable network coding; Minimum completion time, Decoding delay.

I. INTRODUCTION

Network Coding (NC) gained much attention in the past decade after its first introduction in the seminal paper [2]. In the last lustrum, an important subclass network coding, namely the Instantly Decodable Network Coding (IDNC) was an intensive subject of research [3]–[23] thanks to its several benefits, such as the use of simple binary XOR to encode/decode packets, no buffer requirement, and fast progressive decoding of packets. which is much favorable in many applications (e.g. roadside to vehicle safety messages, satellite networks and

Ahmed Douik and Mohamed-Slim Alouini are with Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division at King Abdullah University of Science and Technology (KAUST), Thuwal, Makkah Province, Saudi Arabia, email: {ahmed.douik,slim.alouini}@kaust.edu.sa

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

Tareq Y. Al-Naffouri is with both the CEMSE Division at King Abdullah University of Science and Technology (KAUST), Thuwal, Makkah Province, Saudi Arabia, and the Electrial Engineering Department at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Eastern Province, Saudi Arabia, e-mail: tareq.alnaffouri@kaust.edu.sa.

IPTV) compared to the long buffering time needed in other NC approaches before decoding.

For as long as the research on IDNC has existed, there were two main metrics that were considered in the literature as measures of its quality, namely the *completion time* [8] and the *decoding delay* [3]. The former measures how fast the sender can complete the delivery/recovery of requested packets whereas the latter measures how far the sender is from being able to serve all the unsatisfied receivers in each and every transmission. For quite some time, these two metrics were considered for optimization separately in many works. Though both proved to be NP-hard parameters to minimize, many heuristics has been developed to solve them in many scenarios [3], [6], [8], [10], [20], [22], [23], but again separately. In fact, it can be easily inferred from [8] and [3] that the policies derived so far to optimize one usually degrades the other.

It was not until very recently until one work has aimed to derive a policy that can balance between these two metric and achieve an intermediate performance for both of them [24]. Nonetheless, there does not exist, to the best of our knowledge, any work that aims to explore how these two metrics can be controlled together in order to achieve an even better performance than the currently best known solutions. For instance, every time an unsatisfied user receives a coded packet that is not targeting it, its decoding delay increases and so does its individual completion time. Although this fact was noted for erasure-less transmissions in [24], it was used to strike a balance in performance between both metrics and not to investigate whether a smart control of such decoding delay effects will further reduce the overall completion time compared to its current best achievable performance.

In wireless networks, packet loss occurs due to many phenomena related to the mobility and the propagation environment and they are seen as packet erasures at higher communication layers [25]. This erasure nature of the links affects the delivery of meaningful data and thus affects the ability of users to synchronously decode the information flow. As a consequence a better use of the channel and network does not mean an effective better throughput at higher communication layers [25]. Numerous research has been dedicated to understand the different delay aspects in NC. In our previous work [1], we considered a prompt and perfect reception of the

feedback. This assumption is not realistic in practice because of the impairments in the feedback channel. In this paper, we aim to extend the work in [1] by studying the completion time reduction problem of G-IDNC in the persistent erasure channel (PEC) model on both the forward and backward channels, and in the presence of lossy feedback intermittence.

In this paper, we aim to design a new completion time reduction algorithm through decoding delay control in persistent erasure channels. We first derive more general expressions of the individual and overall completion times over erasure channels as a function of the users' decoding delays. Since finding the optimal schedule of coded packets to minimize the overall completion time is NP-hard [26], we design two greedy heuristic that aims to minimize the probability of increasing the maximum of these decoding-delay-dependent completion time expressions after each transmission. In the first heuristic, we show that this process can be done by partitioning the *IDNC graph* into layers with descending order of user completion time criticality before each transmission. The coding combination for this transmission is then designed by going through these descending order layers sequentially and selecting the combination that minimizes the probability of any decoding delay increments within each layer. This is done while maintaining the instant decodability constraint of the overall coding combination for the targeted users in the more critical layer(s). In the second heuristic, we use a binary optimization algorithm with multi-layer objective function. We, then, extend our study to the limited feedback environment. Finally, we compare through simulations the performance of our designed algorithm to the best known completion time and decoding delay reduction algorithms.

The rest of this paper is divided as follows: In Section II, we present our system model and in Section III we present the channel and feedback model. The problem formulation using the decoding delay is presented in Section IV. Analytic formulation of the sub optimal solution at each transmission is provided in Section V. In Section VI, algorithms to solve the former problem are presented. Section VII presents the extension of the study to the limited feedback scenario. Before concluding in Section IX, simulations results are illustrated in Section VIII.

II. SYSTEM MODEL AND PARAMETERS

The model we consider in this paper consists of a wireless sender that is required to deliver a frame (denoted by $\mathcal{N}$) of N source packets to a set (denoted by $\mathcal{M}$) of M users. Each user is interested in receiving the N packets of $\mathcal{N}$. In an *initial phase*, The sender transmits the N packets of the frame uncoded. Each user listens to all transmitted packets and feeds back to the sender an acknowledgement for each successfully received packet.

After the *initial phase*, two sets of packets are attributed to each user i at the sender:

- The *Has* set (denoted by $\mathcal{H}_i(t)$) is defined as the set of packets successfully received by user i.

- The *Wants* set (denoted by $\mathcal{W}_i(t)$) is defined as the set of packets that are lost by user i. In other words,we have $\mathcal{W}_i = \mathcal{N} \setminus \mathcal{H}_i$.

The BS saves the information obtained after the transmission at time $(t-1)$ in a *feedback matrix* (FM) $\mathbf{F}(t) = [f_{ij}(t)], \ \forall \ i \in \mathcal{M}, \ \forall \ j \in \mathcal{N}, \ \forall \ t > 0$ such that:

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

For ease of notation, we will assume that the time index t denotes the transmission number within the recovery phase and thus $t = 0$ refers to its beginning. Therefore, the sets $\mathcal{H}_i(0), \mathcal{W}_i(0)$ and $\mathcal{U}_i(0)$ refers to the sets at the beginning of the recovery phase (i.e. the sets obtained after the initial transmissions). After this initial transmission, the recovery phase starts at time $t = 1$. In this phase, the BS uses binary XOR to encode the source packets to be send. The packet combination is chosen using the information available in the FM and the expected erasure patterns of the links. In this phase, the transmitted coded packets can be one of the following three options for each user i:

- *Non-innovative:* A packet is non-innovative for user i if it does not bring new useful information. In other words, all the source packets encoded in it were successfully received and decoded previously.

- *Instantly Decodable:* A packet is instantly decodable for user i if the encoded packet contain at most one packet the user do not have so far. In other words, it contains *only one packet* from $\mathcal{W}_i$.

- *Non-Instantly Decodable:* A packet is non instantly decodable for user i if it contains more than one source packet the user do not have so far. In other words, it contains at least two packets from $\mathcal{W}_i$.

After the *initial phase*, the *recovery phase* begins. In this phase, the sender exploits the diversity in received packets at the different users to transmit network coded combinations of the source packets. After each transmission, users update the sender in case they receive the coded packet and decode missing source packets from it. This process is repeated until all users complete the reception of all frame packets.

We define, like in [5], the targeted users by a coded packet (or a transmission) as the users to whom the BS indented the packet combination when encoding it. Given a schedule S of coded packets transmitted by the sender, we define the individual completion time, overall completion time and the decoding delay, like in [3], [10], [12], [20], [24], as follows:

Definition 1. *The individual completion time $\mathcal{S}_i(S)$ of user i is the number of recovery transmissions required until this user obtained all its requested packets.*

Definition 2. *The overall completion time $\mathcal{S}(S)$ of a frame is the number of recovery transmissions required until all users obtain all their requested packets. It easy to infer that $\mathcal{S}(S) = \max_{i \in \mathcal{M}} \mathcal{S}_i(S)$.*
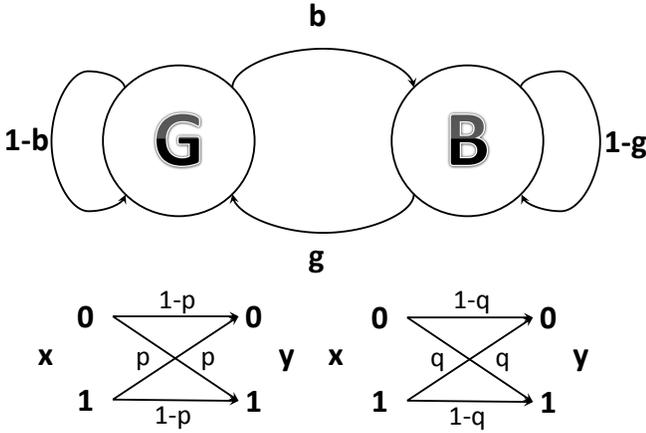
Fig. 1.   The two state Gilbert-Elliott channel.

Definition 3. *At any recovery phase transmission a time t, a user i, with non-empty Wants set, experiences a one unit increase of decoding delay if it successfully receives a packet that is either non-innovative or non-instantly decodable. Consequently, the decoding delay $D_i(S)$ experienced by user i given a schedule S is the number of received coded packets by i before its individual completion, which are non-innovative or non-instantly decodable.*

III. FORWARD CHANNEL MODEL

Following the same model as [6], [17], [23], the forward channel (from the BS to the users) is modeled as a Gilbert-Elliott channel (GEC) [27]. The good and bad states are designed by G and B, respectively. In the original version of the Gilbert-Elliott channel [28], the good state was assumed to be error free and the bad state always in error (i.e. $p = 0$ and $q = 1$ in Figure 1). This zero error probability in the good state allowed the computation in close form of the capacity [29]. This model was then extended to incorporate non zero error probability in the good state. This formulation allow the modeling of multiple fading scenarios [27]. In this paper, this extended channel form will be used (see Figure 1).

Due to the persistent erasure of the Markov chain, the channel has memory that depends on the transition probabilities between the good and bad states. Let C_i^f denote the state of the forward channels. For each user i, the transition probability, from time $t - 1$ to t, are:

$$\mathbb{P}(C_i^f(t) = G | C_i^f(t-1) = B) = g_i^f$$
$$\mathbb{P}(C_i^f(t) = B | C_i^f(t-1) = B) = 1 - g_i^f$$
$$\mathbb{P}(C_i^f(t) = B | C_i^f(t-1) = G) = b_i^f$$
$$\mathbb{P}(C_i^f(t) = G | C_i^f(t-1) = G) = 1 - b_i^f, \qquad (2)$$

where the superscript f indicates the transmission (forward) channel. The use of a superscript (f herein) will be motivated in Section VII when studying the limited feedback scenario.

The probabilities, for user i, for a packet to be erased in the good and bad state are respectively p_i^f and q_i^f.

Since this Markov chain is time-homogeneous (the process can be described by a single, time-independent matrix), we express the probabilities to be in Good or Bad state (steady-state probabilities) for both the transmission and feedback channel, as:

$$\mathcal{P}_{G_i^f} = \mathbb{P}(C_i^f = G) = \frac{g_i^f}{g_i^f + b_i^f}$$
$$\mathcal{P}_{B_i^f} = \mathbb{P}(C_i^f = B) = \frac{b_i^f}{g_i^f + b_i^f}, \qquad (3)$$

We define the memory factor as an indicator of the correlation between the states at different times. A high value of this factor means that the channel is likely to stay in the same state during the following transmissions. In contrast, a small value indicates that the state of the channel changes in an independent manner. For each user i, the memory factor of the forward channel μ_i can be formulated as:

$$\mu_i = 1 - g_i^f - b_i^f \qquad (4)$$

The persistent erasure channel is more likely to stay in the state during the next transmission than switching states. Therefore, we have $0 \le \mu_i \le 1$, $\forall\, i \in \mathcal{M}$. Let $\mu = \dfrac{\sum\limits_{i \in \mathcal{M}} \mu_i}{M}$ be the average memory for the forward link. The memory-less channels are a special case of this persistent erasure channel that can be obtained by setting the memory factor for each user to 0. In other words, by setting $\mu = 0$.

We assume that each user is seen through a channel that is independent from all the other users and thus no correlation exists between the different channels. The state transition probabilities of all the users are know by the BS and used when choosing the packet to be encoded.

IV. PROBLEM FORMULATION USING DECODING-DELAY-DEPENDENT EXPRESSIONS

Let $d_i(\kappa(t))$ be the decoding delay increase for user i, at time t, after the transmission $\kappa(t)$. Define $D_i(t)$ as the total decoding delay experienced by user i until the transmission at time t (i.e. $D_i(t) = \sum\limits_{n=1}^{t} d_i(\kappa(t))$). Let $\mathcal{C}_i(S)$ denote the completion time for user i given a certain schedule S of coded packets. In other words, $\mathcal{C}_i$ is the total number of transmissions required, since the beginning of the *recovery phase*, so that user i successfully receives all its missing packets.

The completion time for the whole session, denoted by $\mathcal{C}$, is the required time, from the beginning of the *recovery phase*, to serve all users. Therefore, $\mathcal{C}$ is controlled by the largest user completion time. In other words, we have:

$$\mathcal{C} = \max_{i \in \mathcal{M}} \mathcal{C}_i. \qquad (5)$$

The following theorem introduces a decoding-delay-dependent expression for the individual completion time of

3

user i and the overall completion time, given the transmission of schedule S from the sender over erasure channels.

Theorem 1. *For a relatively large number of packets N, and a schedule S of transmitted packets transmitted by the sender until overall completion occurs to all users, the individual completion time for user i can be approximated by:*

$$\mathcal{C}_i(S) \approx \frac{|\mathcal{W}_i(0)| + D_i(S) - \alpha_i}{1 - \alpha_i} \qquad (6)$$

where

$$\alpha_i = \frac{g_i^f p_i^f + q_i^f b_i^f}{g_i^f + b_i^f} \qquad (7)$$

Consequently, the overall completion time for the same schedule S can be expressed as:

$$\mathcal{C}(S) \approx \max_{i \in \mathcal{M}} \left\{ \frac{|\mathcal{W}_i(0)| + D_i(S) - \alpha_i}{1 - \alpha_i} \right\} \qquad (8)$$

Proof: The proof can be found in Appendix B. ∎

In the rest of the paper, we will use the approximation with equality as it indeed holds for large N. We can thus formulate the minimum completion time problem as finding the schedule of coded packet S^*, such that:

$$S^* = \arg\min_{S \in \mathcal{S}} \{\mathcal{C}(S)\}$$
$$= \arg\min_{S \in \mathcal{S}} \left\{ \max_{i \in \mathcal{M}} \left\{ \frac{|\mathcal{W}_i(0)| + D_i(S) - \alpha_i}{1 - \alpha_i} \right\} \right\}, \qquad (9)$$

where $\mathcal{S}$ is the set of all possible transmission schedules of coded packets.

Clearly, finding this optimal schedule at time $t = 0$ through the above optimization formulation is very difficult. This is true due to the dynamic nature of erasures and the dependence of the optimal schedule of their effect, which makes the above equations anti-causal (i.e. current result depends on input from the future). Moreover, we know from the literature that optimizing the completion time over the whole *recovery phase* is intractable [20] even for the erasure-less scenario [24]. On the other hand, this formulation shows that the only terms affected by the schedule in the individual and overall completion time expressions are the decoding delay terms of the different users. Consequently, controlling such decoding delays in a smart way throughout the selection of the coded packet schedule can indeed affect the reduction of the completion time significantly. We will thus design a new heuristic algorithm in the next section that takes this fact into consideration. In the rest on this paper, we will refer to the best packet combination that reduce the completion time at each time step as the optimal solution.

V. COMPLETION TIME REDUCTION

A. Critical Criterion

From (9), we can see that the optimal schedule is the one that achieves the minimum overall growth in the individual completion time expressions in (6) $\forall\ i \in \mathcal{M}$. Since we know that finding such schedule for the entire recovery phase, prior

to its start, is intractable, we will design our heuristic algorithm such that, in each transmission time $t > 0$, it minimizes the probability of increase of the maximum of such expressions over all users compared to their state before this transmission. To formally express this criterion, let us first define $\mathcal{C}_i(t)$ as:

$$\mathcal{C}_i(t) = \frac{|\mathcal{W}_i(0)| + D_i(t) - \alpha_i}{1 - \alpha_i} \qquad (10)$$

In other words, $C_i(t)$ is the anticipated individual completion time of user i if it experiences no further decoding delay increments starting from time t. Thus, the philosophy of our proposed heuristic algorithm is to transmit the coded packet $\kappa(t)$ at time t such that:

$$\kappa^*(t)$$
$$= \arg\min_{\kappa(t) \in \mathcal{G}(t)} \left\{ \mathbb{P}\left(\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\} > \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\} \right) \right\}$$
$$\qquad (11)$$

We will refer to (11) as the critical criterion. Let $\mathcal{P}(t)$ be the set of users that can potentially increase $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\}$ at time t compared to $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\}$ if they are not targeted by $\kappa(t)$. The set can be mathematically defined as follows:

$$\mathcal{P}(t) = \left\{ i \in \mathcal{M} \ \middle| \ \frac{|\mathcal{W}_i(0)| + (D_i(t-1) + 1) - \alpha_i}{1 - \alpha_i} \right.$$
$$\left. > \frac{|\mathcal{W}_j(0)| + D_j(t-1) - \alpha_j}{1 - \alpha_j} \right\}, \qquad (12)$$

where $j = \arg\max_{k \in \mathcal{M}} \left\{ \frac{|\mathcal{W}_k(0)| + D_k(t-1) - \alpha_k}{1 - \alpha_k} \right\}$. We will refer to this set as the "highly critical set".

B. Optimization Problem

Let $e_i(t)$ be the probability that user i loose the transmission at time t. This formulation of this probability is:

$$e_i(t) = \begin{cases} p_i & \text{if } C_i^f(t) = G \\ q_i & \text{otherwise} \end{cases} \qquad (13)$$

Define $\tau(\kappa(t))$ as the set of users that are targeted by the transmission $\kappa(t)$. The following theorem defines a maximum weight clique algorithm that can satisfy the critical criterion.

Theorem 2. *The critical criterion in (11) can be achieved by selecting $\kappa^*(t)$ according to the following optimization problem:*

$$\kappa^*(t) = \underset{\kappa(t) \in \mathcal{G}}{argmax} \left\{ \sum_{i \in (\mathcal{P}(t) \cap \tau)} log\left(\frac{1}{e_i(t)} \right) \right\}. \qquad (14)$$

In other words, the transmission $\kappa(t)$ than can satisfy the critical criterion can be selected using a maximum weight clique problem in which the weight of each vertex v_{ij} in $\mathcal{P}(t)$

can be expressed as:

$$w_{ij}^* = log\left(\frac{1}{e_i(t)}\right). \qquad (15)$$

Proof: The proof can be found in [1]. ∎

VI. Proposed Heuristics Algorithms

A. Maximum Weight Clique Solution

1) Instantly Decodable Network Coding Graph: To look for possible packet combinations the base station can make, we use the G-IDNC graph representation. To construct this G-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we first create a vertex $v_{ij} \in \mathcal{V}$ for each packet $j \notin \mathcal{H}_i, \ \forall \ i \in \mathcal{M}$. We then connect with an edge all v_{ij} and v_{kl} in $\mathcal{V}$ if one of the two following conditions is true:

- $j = l \Rightarrow$ Packet j is needed by both clients i and k.
- $j \in \mathcal{H}_k$ and $l \in \mathcal{H}_i \Rightarrow$ The packet combination $j \oplus l$ is instantly decodable for both clients i and k.

Given this graph formulation and according to the analysis done in [26], the set of all packet feasible combinations in G-IDNC is represented by all maximal cliques in $\mathcal{G}$. To generate a packet combination, binary XOR is applied to all the packets identified by the vertices of a selected maximal clique κ in $\mathcal{G}$. The targeted clients by this transmission κ are those identified by the vertices of the selected maximal clique.

2) Multi-layer Greedy Algorithm: Despite the importance of the satisfaction of the critical criterion in order to minimize the probability of increase of the maximum individual completion time, it may not fully exploit the power of IDNC. In other words, once a clique is chosen according to (14) from among the users in the highly critical set $\mathcal{P}(t)$, there may exist vertices belonging to other users that can form an even bigger clique. Thus, adding this vertex to the clique and serving this user will benefit them without affecting the IDNC constraint for the users belonging to $\mathcal{P}(t)$.

To schedule such vertices and their users, let $\mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_h$ (with $h \in \mathbb{N}$) be the sets of vertices of $\mathcal{G}(t)$, such that $v_{ik} \in \mathcal{G}_n$ if the following conditions are true:

- $\mathcal{C}_i(t-1) + \dfrac{n}{1-\alpha_i} > \mathcal{C}_j(t-1)$.
- $\mathcal{C}_i(t-1) + \dfrac{n-1}{1-\alpha_i} \le \mathcal{C}_j(t-1)$.

where $j = \underset{i \in \mathcal{M}}{\mathrm{argmax}} \{\mathcal{C}_i(t-1)\}$. Consequently, the IDNC graph at time t is partitioned into h layers with descending order of criticality. By examining the above condition, the vertices of the users of $\mathcal{P}(t)$ are all in layer $\mathcal{G}_1$. Moreover, the n-th layer of the graph includes the vertices of the users who may eventually increase $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t+n)\}$ if they experience n decoding delay increments in the subsequent n transmissions. Consequently, a user with vertices belonging to $\mathcal{G}_i$ is more critical than another with vertices belonging to $\mathcal{G}_j$, $j > i$, as the former has a higher chance to increase the overall completion time.

In order to guarantee the satisfaction of the critical criterion, our proposed algorithm first finds the maximum weight clique

---

Algorithm 1 Maximum Weight Vertex Search Algorithm

Require: $\mathbf{F}$, p_i and $C_i(t-1)$, $\forall \ i \in \mathcal{M}$.

  Initialize $\kappa^* = \varnothing$.
  Construct $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1), \mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2), ..., \mathcal{G}_h(\mathcal{V}_h, \mathcal{E}_h)$.
  for l=1 **to** h **do**
    $\mathcal{G} \leftarrow \mathcal{G}_l$.
    for all $v \in \kappa^*$ **do**
      Sets $\mathcal{G} \leftarrow \mathcal{G}(\kappa^*)$.
    end for
    while $\mathcal{G} \neq \varnothing$ **do**
      Compute w_{ij}^* and w_{ij} using (15) and (16).
      Select $v^* = \underset{v_{ij} \in \mathcal{G}}{\mathrm{argmax}} \{w_{ij}\}$.
      Sets $\kappa^* \leftarrow \kappa^* \cup v^*$.
      Sets $\mathcal{G} \leftarrow \mathcal{G}(\kappa^*)$.
    end while
  end for

κ^* in layer $\mathcal{G}_1$ as mandated by Theorem 2. We then construct $\mathcal{G}_2(\kappa^*)$ including each vertex in $\mathcal{G}_2$ that is adjacent to all vertices in κ^* (i.e. forms a bigger clique with κ^*). After assigning the same weights defined in (15), the maximal weight clique in $\mathcal{G}_2(\kappa^*)$ is found and added to κ^*. This process is repeated for each layer $\mathcal{G}_i, i \le h$ of the graph to find the selected maximal weight clique $\kappa^* \in \mathcal{G}(t)$ to be transmitted at time t.

Since finding the maximum weight clique in the G-IDNC graph is NP-hard [26] we will use the same simple vertex search approach with modified weights introduced in [19] after tailoring the weights to the ones defined in (15). Let w_{ij} be the modified weights, which can be expressed as:

$$w_{ij} = (w_{ij}^* + 1) \times \sum_{v_{kl} \in \nu(v_{ij})} w_{kl}^*, \qquad (16)$$

where $\nu(v_{ij})$ is the set of adjacent vertices of v_{ij} within its layer.

B. Binary Particle Swarm Optimization Solution

Particle swarm Optimization (PSO) is a population based search algorithm based on the simulation of the social behavior of animals. It was introduced in [30]–[32] by Eberhart and Kennedy for continuous functions. To each individual in the multidimensional space is associated two vectors: the position vector and the velocity vector. The velocity vector determines in which direction the position vector should evolve.

In [33], the authors extended their algorithm to the binary optimization and in [34], Khanesar proposed a novel binary PSO. This algorithm is based on a new definition for the velocity vector of binary PSO. In order to state the outlines of this algorithm, we first introduce the following quantities: Let X_i be the position of particle i, P_{ibest} is the best position particle i visited and P_{gbest} the best position visited by any particle. Let L be the total number of particles and T the number of iteration of the BPSO algorithm.

Let V_{ij}^1 and V_{ij}^0 be the probabilities that a bit j of the particle

i changes from 0 to 1 and from 1 to 0, respectively. These probabilities are computed using the following expressions:

$$V_{ij}^1 = wV_{ij}^1 + d_{ij,1}^1 + d_{ij,2}^1 \tag{17}$$

$$V_{ij}^0 = wV_{ij}^0 + d_{ij,1}^0 + d_{ij,2}^0, \tag{18}$$

where w is a random number in the range $[-1, 1]$ chosen at the beginning of the BPSO algorithm and $d_{ij,1}$, $d_{ij,2}$, $d_{ij,1}^0$, and $d_{ij,2}^0$ are extracted using the rule below:

If $P_{ibest}^j = 1$ Then $d_{ij,1}^1 = c_1r_1$ and $d_{ij,1}^0 = -c_1r_1$

If $P_{ibest}^j = 0$ Then $d_{ij,1}^0 = c_1r_1$ and $d_{ij,1}^1 = -c_1r_1$

If $P_{ibest}^j = 1$ Then $d_{ij,2}^1 = c_2r_2$ and $d_{ij,2}^0 = -c_2r_2$

If $P_{ibest}^j = 1$ Then $d_{ij,2}^0 = c_2r_2$ and $d_{ij,2}^0 = -c_2r_2 \tag{19}$

where c_1 and c_2 are fixed factor determined by user and r_1 and r_2 are two random variable in the range $[0, 1]$ chosen at each iteration of the BPSO algorithm. The velocity V_{ij}^c of bit j of particle i is defined as:

$$V_{ij}^c = \begin{cases} V_{ij}^1 \text{ if } x_{ij} = 0 \\ V_{ij}^0 \text{ if } x_{ij} = 1 \end{cases} \tag{20}$$

The normalized velocity is obtained using the sigmoid function. This function is defined as follows:

$$\text{sig}(x) = \frac{1}{1 + e^{-x}} \tag{21}$$

Let V_{ij}' be the normalized velocity (i.e. $V_{ij}' = \text{sig}(V_{ij}^c)$). The position update of the bit j of particle i is computed as follows:

$$x_{ij}(t+1) = \begin{cases} \overline{x}_{ij}(t) \text{if} & r_{ij} < V_{ij}' \\ x_{ij}(t) \text{if} & r_{ij} > V_{ij}' \end{cases} \tag{22}$$

where $\overline{x}$ is the binary complementary of x and r_{ij} is a random number in the range $[0, 1]$ chosen at each iteration and at each bit of the particle.

Let $\mathcal{F}$ be the function of interest that we want to minimize. The outline of the BPSO algorithm are the following:

1) Initialize the swarm X_i, the position of particles are randomly initialized within the hyper-cube.
2) Evaluate the performance $\mathcal{F}$ of each particle, using its current position $X_i(t)$.
3) compare the performance of each individual to its best performance so far: if $(\mathcal{F}(X_i(t)) < \mathcal{F}(P_{ibest}))$:
   $\mathcal{F}(P_{ibest}) = \mathcal{F}(X_i(t))$
   $P_{ibest} = X_i(t)$
4) Compare the performance of each particle to the global best particle:if $(\mathcal{F}(X_i(t)) < \mathcal{F}(P_{gbest}))$:
   $\mathcal{F}(P_{gbest}) = \mathcal{F}(X_i(t))$
   $P_{gbest} = X_i(t)$
5) change the velocity of the particle, V_i^0 and V_i^1.
6) Calculate the velocity of change of the bits, V_i^c.
7) Generate the random variable r_{ij} in the range: (0,1). Move each particle to a new position.
8) Go to step 2, and repeat $(T-1)$ times.

Let $\phi_i(\kappa, t)$ be the weight of user i when sending the packet combination κ at time t. Following the result of the previous section, $f_i(\kappa, t)$ is expressed as:

$$\phi_i(\kappa, t) = \tag{23}$$
$$\begin{cases} \log\left(\dfrac{1}{e_i(t)}\right) & \text{if } i \in \tau \cap \mathcal{P}(t) \\ 0 & \text{otherwise} \end{cases}$$

As for the multi-layer solution in the maximum weight clique problem, we want to include users that are not in the critical layer under the condition that their inclusion to not disturb the instantaneous decodability of the targeted users in the critical layer. This inclusion is done using layer prioritization i.e. the inclusion of a user in layer $\mathcal{P}_m$ should not bother the instant dependability of users in layer $\mathcal{P}_n, n > m$. To reproduce this concept of prioritization using a single real function, we use the sigmoid function. The new objective function to maximize is the following:

$$\phi_i'(\kappa, t) = \sum_{i \in M_w)} \text{sig}(\tilde{\phi}_i(\kappa, t)) + M * (h - P(i)) \tag{24}$$

where h is the total number of layers, $P(i)$ is the index of the layer of user i and $\tilde{\phi}_i$ defined as follows:

$$\tilde{\phi}_i(\kappa, t) = \log\left(\frac{1}{e_i(t)}\right) \tag{25}$$

The sigmoid function is an increasing function between $[0, 1]$ and therefore the utility of each user is shifted according to the number of the layer he is in. Since no more that M players can be simultaneously on the same layer, then all these layers are not overlapping and a single element of one layer will yield a better objective that all the sum of all the objective of user in layer below him. This function respects the prioritization as the multi-layer graph. The function of interest to minimize , at each time instant t, can be written as $\mathcal{F}(\kappa) = \phi_i'(\kappa, t)$. The following theorem gives the convergence of the whole system in our cases:

Theorem 3. *For a number of particle equal to the number of packets (i.e. $L = N$) and a sparse initialisation vector, the overall system will converge*

Proof: The proof of this Theorem can be found in Appendix C. ∎

VII. EXTENSION TO THE LIMITED FEEDBACK SCENARIO

In this section, we extend our previous analysis to the limited feedback scenario. First, we introduce the system model, the backward channel model and the feedback protocol. We, then, derive the expression of the optimal packet combination to reduce the completion time in such lossy feedback environment. Finally, we propose modified version of previously introduced algorithm to effectively reduce the completion time.

A. System Model

Due to the feedback loss that can occur in the system, at the end of the *initial phase*, three sets of packets are attributed to each user i at the sender:

- The *Has* set (denoted by $\mathcal{H}_i(t)$) is defined as the set of packets successfully received by user i.
- The *Wants* set (denoted by $\mathcal{W}_i(t)$) is defined as the set of packets that are lost by user i. In other words,we have $\mathcal{W}_i = \mathcal{N} \setminus \mathcal{H}_i$.
- The Uncertain set (denoted by $\mathcal{U}_i(t)$). It is the sets of packets which the BS is not certain if either the packet or the feedback were erased. We have $\mathcal{U}_i(t) \subseteq \mathcal{W}_i(t)$.

The BS saves the information obtained after the transmission at time $(t - 1)$ in a *feedback matrix* (FM) $\mathbf{F}(t) = [f_{ij}(t)]$, $\forall\, i \in \mathcal{M}$, $\forall\, j \in \mathcal{N}$, $\forall\, t > 0$ such that:

$$
f_{ij}(t) = \begin{cases} 0 & \text{if } j \in \mathcal{H}_i(t) \\ 1 & \text{if } j \in \mathcal{W}_i(t) \setminus \mathcal{U}_i(t) \\ x & \text{if } j \in \mathcal{W}_i(t) \cap \mathcal{U}_i(t). \end{cases} \tag{26}
$$

B. Backward Channel Model and Feedback Protocol

We model the backward channel (from the users to the BS), as for the forward channel, by a Gilbert-Elliott channel. The parameters of the backward channel are defined in the same way as those of the forward channel with a superscript b. For example, C_i^b will denote the state of the forward channels. For each user i, the memory factor of the backward channel ψ_i and the average memory can be formulated as:

$$
\psi_i = 1 - g_i^b - b_i^b
$$

$$
\psi = \frac{\sum\limits_{i \in \mathcal{M}} \psi}{M}. \tag{27}
$$

When both the forward and the backward channel have the same transitions probabilities (i.e. $g_i^f = g_i^b$ and $b_i^f = b_i^b, \forall\, i \in \mathcal{M}$), the channel is said to be identically distributed and when they are experiencing the same channel realization (i.e $C_i^f(t) = C_i^b(t), \forall\, t > 0$) the channel is said to be reciprocal.

Before transmitting the data packet, the BS first transmits N bits representing the source packets that are combined in the encoded packet (i.e. if the packet is in the combination the corresponding bit is set to 1 otherwise it is set to 0). Each user, after listening to the header, continues to listen if the packet combination is instantly decodable for him. Otherwise, energy is saved by ignoring the next transmission. The targeted users by a transmission are the users that can instantly decode a packet from that transmission.

Time is divided into frames of same length equal to T_f time slots. Each of these frame is composed of a downlink sub-frame and an uplink sub-frame. The length of the downlink and uplink sub-frame are respectively T_d and T_u. In other words, we have $T_f = T_d + T_u$. The BS transmits the packet combinations during the downlink sub-frame and do not get any feedback the meanwhile. During the uplink sub-frame, the BS listens to the feedbacks sent from the different users and do not transmit packets. After each downlink sub-frame, users

that received and managed to decode the packet combination during that downlink sub-frame acknowledges its reception by sending a feedback during the uplink sub-frame. Define T_{u_i} as the time-slot in the uplink sub-frame in which user i is able to send feedback (i.e. $1 \le T_{u_i} \le T_u$, $\forall\, i \in \mathcal{M}$). In other words, user i can transmit acknowledgement, during frame number n, at time $t = nT_f - T_u + T_{u_i}$. Only targeted users during the downlink sub-frame will send acknowledgement. In other words, if a feedback from one of the targeted user is lost, the BS will not get any feedback from this user until the next transmission in which it is targeted. Each feedback sent from a user consists of N bits indicating all previously received/lost packets. As a consequence, after receiving feedback from an arbitrary user, the states of all its packet in the FM will be certain (i.e. $\mathcal{U}_i = \varnothing$). The BS uses the feedback to update the feedback matrix. This process is repeated until all users report that they obtained all the wanted packets.

In order to be able to accurately estimate the state of the forward/backward channels for each user, we impose, as in [23], that for each targeted user, from the last time a feedback is heard from that user, there will be at least one packet which is attempted only once. This constraint becomes unnecessary for user that still need only a single source packet. An additional $log_2(T_d)$ are send with the feedback indication the decoding delay encountered by the user during the downlink sub-frame.

C. Transmission/Feedback Loss Probabilities at Time t

In this section, we compute the probability to loose the transmission $e_i(t) \triangleq \mathbb{P}(C_i^p(t) = B)$, $\forall\, i \in \mathcal{M}$ and to loose the feedback $f_i(t) \triangleq \mathbb{P}(C_i^q(t) = B)$, $\forall\, i \in \mathcal{M}$, at time slot t.

In order to compute these probabilities, we first introduce the following variables: Let $n_i^{(-1)}$ and $n_i^{(0)}$ ($n_i^{(-1)} < n_i^{(0)}$) be the indices of the most recent frame where the sender heard a feedback from user i. Let $\lambda_{ij}(n)$ be the set of the time indices when packet j was attempted to user i during frame number n. Define $j_i^{(0)}$ as the last sent packet among those which were attempted only once between the two frames $(n_i^{(-1)} + 1)$ and $n_i^{(0)}$ to user i. This variable can be mathematically defined as:

$$
j_i^{(0)} = \underset{j \in \mathcal{W}_i(n_i^{(0)} \times T_f)}{\arg\max} \; \bigcup_{k=n_i^{(-1)}+1}^{n_i^{(0)}} \lambda_{ij}(k)
$$

$$
\text{subject to } \left| \bigcup_{k=n_i^{(-1)}+1}^{n_i^{(0)}} \lambda_{ij}(k) \right| = 1, \tag{28}
$$

where $\cup_{x \in X} A_x$ is the union of the sets A_x, $\forall\, x \in X$. Let $t_i^{(0)}$ be the time where packet $j_i^{(0)}$ was attempted to receiver i and t_i^* the last time the a feedback was heard from user i.

In other words:

$$t_i^{(0)} = \bigcup_{k=n_i^{(-1)}+1}^{n_i^{(0)}} \lambda_{ij_i^{(0)}}(k) \qquad (29)$$

$$t_i^* = n_i^{(0)} \times T_f - T_u + T_{u_i}. \qquad (30)$$

Given these definitions, we can introduce the following theorem regarding the loss probabilities of the forward $e_i(t)$ and feedback $f_i(t)$ transmissions at any given time t.

Theorem 4. *The probabilities $e_i(t)$ of loosing a transmission from receiver i at time $t > t_i^*$ can be expressed as:*

$$e_i = \qquad (31)$$

$$\begin{cases} \dfrac{p_i^f g_i^f}{p_i^f g_i^f + q_i^f b_i^f}(p_i^f + (q_i^f - p_i^f)b_i^f \sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\ \quad + \dfrac{q_i^f b_i^f}{p_i^f g_i^f + q_i^f b_i^f}(q_i^f + (p_i^f - q_i^f)g \sum_{i=0}^{t-n^0-1} \mu^i) \\ \qquad \text{if } f_{ij^0} = 1 \\ \dfrac{(1-p_i^f)g_i^f}{(1-p_i^f)g_i^f + (1-q_i^f)b_i^f} \\ \qquad \times (p_i^f + (q_i^f - p_i^f)b_i^f \sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\ \quad + \dfrac{(1-q_i^f)b_i^f}{(1-p_i^f)g_i^f + (1-q_i^f)b_i^f} \\ \qquad \times (q_i^f + (p_i^f - q_i^f)g_i^f \sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\ \qquad \text{if } f_{ij^0} = 0 \end{cases}$$

The probabilities $f_i(t)$ of loosing a feedback from user i at time $t > t_i^*$ can be expressed as:

$$f_i = \dfrac{(1-p_i^b)g_i^b}{(1-p_i^b)g_i^b + (1-q_i^b)b_i^b}(p_i^b + (q_i^b - p_i^b)b_i^b \sum_{i=0}^{t-t_i^*-1} \psi^i)$$

$$+ \dfrac{(1-q_i^b)b_i^b}{(1-p_i^b)g_i^b + (1-q_i^b)b_i^b}(q_i^b + (p_i^b - q_i^b)g_i^b \sum_{i=0}^{t-t_i^*-1} \psi^i) \qquad (32)$$

Proof: The proof can be found in Appendix D. ∎

D. Problem Formulation

In order to state the optimization problem we first define this two probability :

- The innovative probability $p_{i,n}(j,t)$: probability that packet j is innovative for user i at time t.
- The finish probability $p_{i,f}(t)$: probability that user i successfully received all his primary packets but $\mathcal{W}_i(t) \neq \varnothing$ at time t.

The expressions of these probabilities is available in the Appendix E.

The following theorem defines a maximum weight clique algorithm that can satisfy the critical criterion.

Theorem 5. *The critical criterion in (11) can be achieved by selecting $\kappa^*(t)$ according to the following optimization problem:*

$$\kappa^*(t)$$

$$= \underset{\kappa(t) \in \mathcal{G}}{argmax} \left\{ \sum_{i \in (\mathcal{P}(t) \cap \tau)} log \left(1 + \dfrac{p_{i,n}(\kappa_i, t)}{\dfrac{e_i(t)}{1-e_i(t)} + p_{i,f}(t)} \right) \right\}. \qquad (33)$$

In other words, the transmission $\kappa(t)$ than can satisfy the critical criterion can be selected using a maximum weight clique problem in which the weight of each vertex v_{ij} in $\mathcal{P}(t)$ can be expressed as:

$$w_{ij}^* = log \left(1 + \dfrac{p_{i,n}(j,t)}{\dfrac{e_i(t)}{1-e_i(t)} + p_{i,f}(t)} \right). \qquad (34)$$

Proof: The proof can be found in Appendix F. ∎

E. Proposed Algorithm

1) Maximum Weight Clique Solution: In order to minimize the completion time in G-IDNC, we look for all possible combinations of source packets then take the combination of these packets that guarantees the minimum delay for this transmission. To represent all the feasible packet combinations, we use the graph model introduced in [5] and called the lossy D-IDNC graph (LG-IDNC). To construct the LG-IDNC graph, we first introduce the expected decoding delay increase $d_{ij,kl}(j \oplus l)$ for two distinct arbitrary users i and k after sending the packet combination $j \oplus l$:

$$d_{ij,kl}(j \oplus l) = \qquad (35)$$
$$(1-e_i)(p_{n,i}(j)p_{n,i}(l) + (1-p_{n,i}(j)(1-p_{n,i}(l)))\overline{p}_{f,i} +$$
$$(1-e_k)(p_{n,k}(j)p_{n,k}(l) + (1-p_{n,k}(j)(1-p_{n,k}(l)))\overline{p}_{f,k},$$

where $\overline{p}_{f,i} = 1 - p_{f,i}$.

To obtain the expected decoding delay increase $d_{ij,kl}(j)$ for these users after sending packet j, we replace l by 0 in (35) and take $p_{n,i}(0) = 0, \forall i \in \mathcal{M}$. To construct the LG-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we first create a vertex $v_{ij}, \forall i \in \mathcal{M}, \forall j \in \mathcal{W}_i$. We then connect two users, if they either need the same packet or if the expected decoding delay increase is lower when sending the packet combination than when sending only one packet. In other words, we connect by an edge two vertices v_{ij} and v_{kl} if one of the following conditions is true:

- C1: $j = l \Rightarrow$ Packet j is needed by both the users i and k.
- C2: $d_{ij,kl}(j \oplus l) \leq min(d_{ij,kl}(j), d_{ij,kl}(l)) \Rightarrow$ The packet combination $j \oplus l$ guarantees a lower decoding delay to the users i and k than packets j and l individually.

Unlike condition C1 that does not require packet combination, C2 involves the combination of packet j and l. It was shown in [26] that in the perfect feedback scenario, all possible

packet combinations is equivalent to a maximal weight clique in the LG-IDNC graph. Therefore the combination that reduces the best the completion time for the current transmission is the maximum weight clique in the LG-IDNC graph. This result can be extended to the lossy intermittent feedback. The BS generate the encoded packet by taking the binary XOR of packet represented by the maximal weight clique in the LG-IDNC graph. Users targeted by this transmission are those represented by this maximal weight clique.

After definition of the LG-IDNC graph, we use the multi-layer algorithm developed in the previous Section VI with the new weights developed in (34).

2) BPSO Algorithm: The same line of thinking used in Section VI applies in the case of the limited feedback scenario. We objective function is defined using the new weights (34) to reflect the uncertainties in the system. The new objective function to maximize is the following:

$$\phi'_i(\kappa, t) = \sum_{i \in M_w)} \text{sig}(\tilde{\phi}_i(\kappa, t)) + M * (h - P(i)) \quad (36)$$

where h is the total number of layers, $P(i)$ is the index of the layer of user i and $\tilde{\phi}_i$ defined as follows:

$$\tilde{\phi}_i(\kappa, t) = \log \left(1 + \frac{p_{i,n}(\kappa, t)}{\dfrac{e_i(t)}{1 - e_i(t)} + p_{i,f}(t)} \right) \quad (37)$$

F. Blind Graph Policies Solution

In the lossy intermittent feedback, uncertainties about the reception state of the different packets and the decodability conditions make the algorithm proposed in [35] non effective to actually reduce the completion time. To solve this problem, we introduce three partially blind algorithms that estimate all the uncertain packets with a predefined policy, update the graph accordingly and finally perform packet selection using the algorithm proposed in [35]. These graph update approaches are the generalization of the update methods proposed in [36] in the context of reducing the completion time with lossy feedback.

1) Pessimist Graph Update: In this approach, all packets that are not fed back by users are considered erased rather than assuming that their feedback is erased. Reconsidering these packets in the following transmissions gives them a greater chance to be reattempted rapidly. Since no acknowledgement is expected to be heard during the downlink sub frame, packets attempted meanwhile are systematically not reconsidered in the following transmissions. If a feedback is heard in the uplink sub frame, the state of the user is updated. Otherwise, all the uncertain packet of that user are reconsidered.

In the pessimist graph update approach, uncertain vertices are removed from the graph during the downlink sub frame and reconsidered in the uplink sub frame if no acknowledgement is heard from the concerned user.

2) Optimist Graph Update: In this approach, all packets that are not fed back by users are considered received and
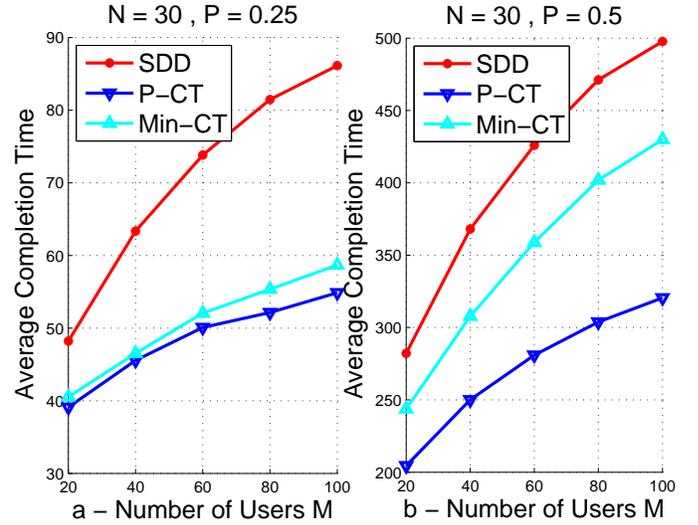

Fig. 2. Mean completion time for G-IDNC versus number of users M.

their corresponding feedback erased. Not reconsidering these packets in the following transmissions gives a greater chance to non attempted packets to be transmitted. Since no feedback can be heard from a user having all its packets in an uncertain state, unless this user is targeted. therefore, users with full uncertain Wants set are reconsidered after the uplink sub frame.

In the optimist graph update approach, uncertain vertices are removed from the graph and reconsidered in the uplink sub frame if the user have full uncertain Wants set.

3) Realistic Graph Update: In this approach, all packets that are not fed back by users are probabilistically considered received and their acknowledgement erased and reciprocally. This approach tends to stochastically balance between reattempting packets with unheard feedback and transmitting new packets. Since during the downlink sub frame no acknowledgement is expected to be heard, then packets are reconsidered with probability $\mathcal{P}_{B_i^f}$ and discarded with probability $\mathcal{P}_{G_i^f}$. In the uplink frame, all the uncertain packets are reconsidered with probability $\mathcal{P}_{B_i^b}$ and removed with probability $\mathcal{P}_{G_i^b}$.

In the realist graph update approach, uncertain vertices are removed from the graph with probability $\mathcal{P}_{G_i^f}$ in the downlink frame and with probability $\mathcal{P}_{G_i^b}$ in the uplink frame.

VIII. SIMULATION RESULTS

In this section, we first present the simulation results comparing the completion time and the decoding delay achieved by the different policies to optimize each in perfect feedback and independent erasure channels. We, then, present the completion time achieved by our policy against the blind policies for a lossy feedback and persistent erasure channels. In the first part, we compare, through extensive simulations the sum decoding delay (denoted by SDD) and the completion time achieved by [8] (denoted by Min-CT) and the completion time achieved by our algorithm (denoted by P-CT) for perfect feedback while using the policy to reduce the sum decoding
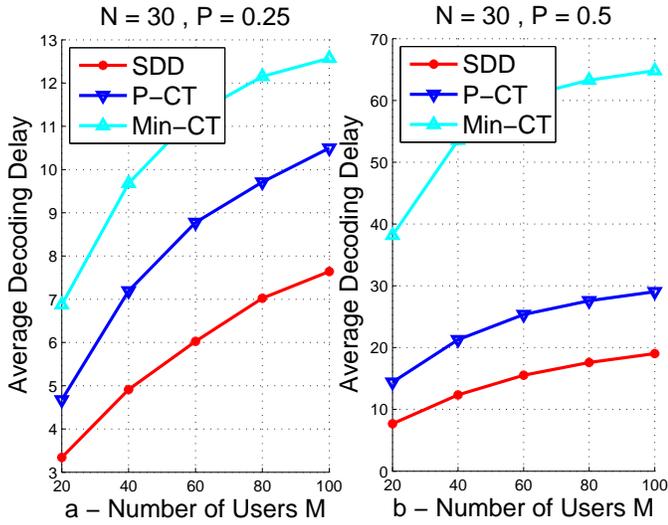
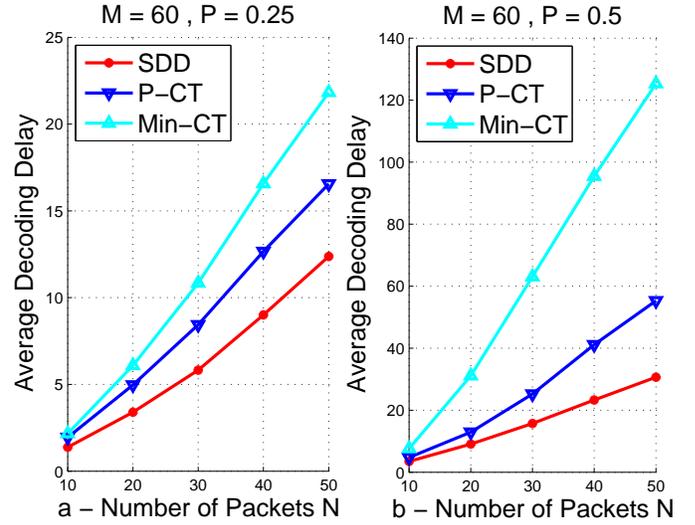Fig. 3. Mean decoding delay for G-IDNC versus number of users M.

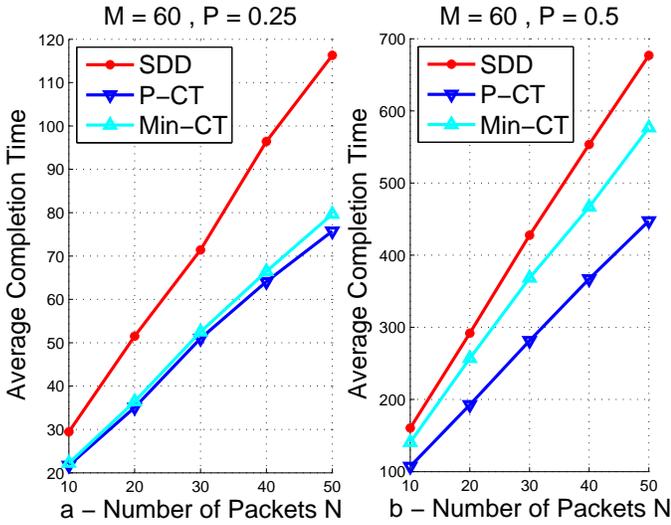Fig. 5. Mean decoding delay for G-IDNC versus number of packets N.

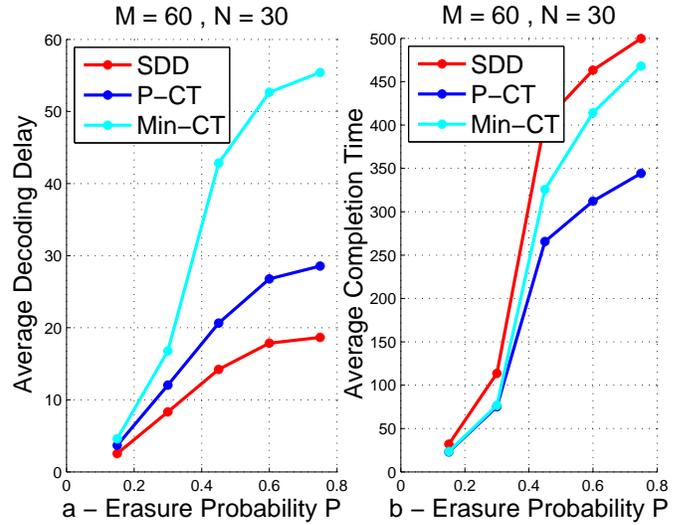Fig. 4. Mean completion time for G-IDNC versus number of packets N.

Fig. 6. Mean delays for G-IDNC versus packet erasure probability P.

delay [3] and the policy [8] and our policy to reduce the completion time [5]. In the second part, we first compare our two heuristics to reduce the completion time for perfect feedback and persistent erasure channels. We, then, compare the completion time achieved by our policy and the blind policies in lossy feedback environment.

In all the simulations, the different delays are computed by frame then averaged over a large number of iterations. We assume that the packet and the feedback erasure probability of all the users change from frame to frame while the average packet erasure probability remain constant. We further assume the symmetric channels for both the forward and backward links. In other words, the erasure probability on the forward and the backward link are the same (the probabilities only and not the channel realizations).

Figure 2 depicts the comparison of the mean completion achieved by the policy to reduce the sum decoding delay

(SDD), [8] policy and our one to reduce the completion time (Min-CT and P-CT) for a perfect feedback and independent erasure channels against M for $N = 60$ and $P = 0.25$ and $P = 0.5$ receptively, where P refers to the average packet erasure probability in the independent erasure channels. Figure 3 illustrates the comparison of the decoding delay for the same inputs. Figure 4 and Figure 5 depicts the comparison of the aforementioned delay aspects against N for $M = 60$ and $P = 0.25$ and $P = 0.5$ receptively and Figure 6 illustrates this comparison against the erasure probability P for $M = 60$ and $N = 30$.

Figure 7 illustrate the comparison of the completion time achieved by the one layered algorithm (denoted by -Graph) and the BPSO algorithm (denoted by -BPSO) for both our policy to reduce the completion time [1] using the decoding delay control (denoted by DDC) and the [8] policy using the shortest stochastic path (denoted by SSP) for a perfect feedback and
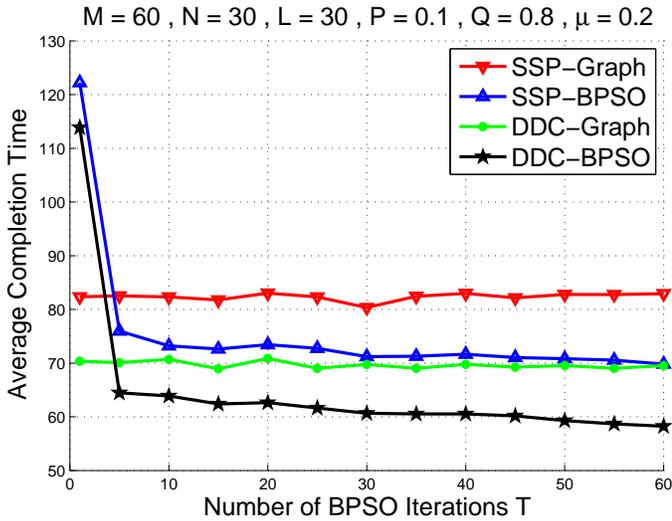
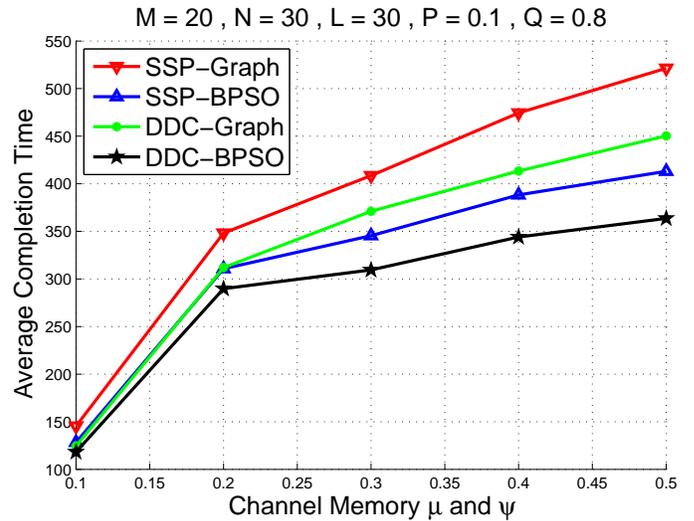Fig. 7. Average completion time versus the number of iteration of BPSO T.

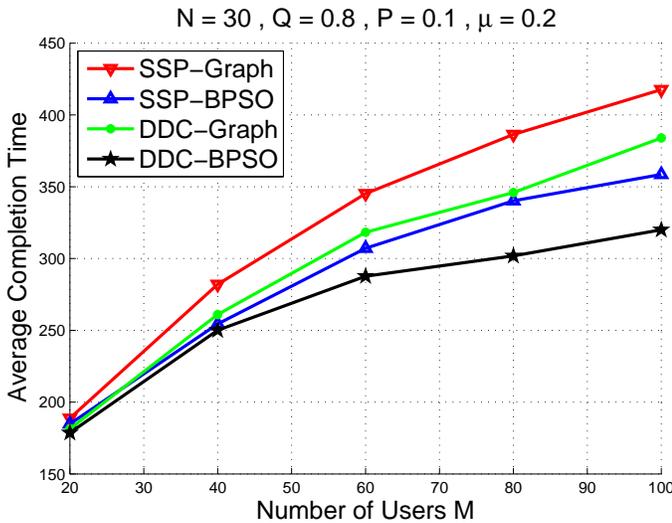Fig. 9. Average completion time versus chanel memory μ and ψ.

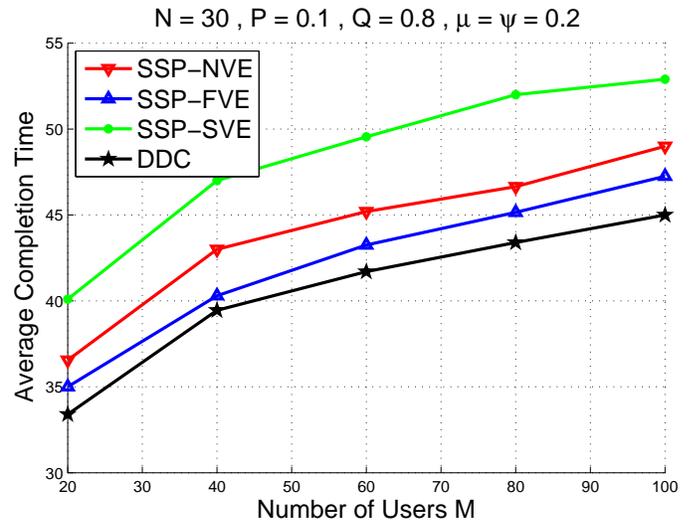Fig. 8. Average completion time versus number of users M.

Fig. 10. Average completion time versus the number of users M.

persistent erasure channels against the number of iteration T for $M = 60$, $N = 30$, $L = 30$, $P = 0.1$, $Q = 0.8$ and $\mu = 0.2$. Figure 8 and Figure 9 depicts the same comparison against the number of users and the memory of the channel respectively for $\mu = 0.2(M = 60)$, $N = 30$, $L = 30$, $P = 0.1$ and $Q = 0.8$.

Figure 10 shows the comparison between the different blind algorithm (denoted by NVE for the pessimist, FVE of the optimist and SVE for the stochastic policy) when using the original formulation of the weights proposed in [8] and our multi-layer graph algorithm using the decoding delay control (denoted by DDC) against the number of users M for $N = 30$, $P = 0.1$, $Q = 0.8$, and $\mu = 0.2$. Figure 11 depicts the same comparison against the channel memory μ and ψ for $M = 60$, $N = 30$, $P = 0.1$, and $Q = 0.8$.

From all the figures, we can clearly see that our proposed completion time algorithm outperforms the completion time

policy proposed in [8] and the blind policies. Moreover it gives the best agreement among the sum decoding delay and the completion in G-IDNC. The completion time policy offers, in average, the minimum sum of all the delay aspects in all situations.

Figure 2.a and Figure 4.a depicts the completion time when applying the sum decoding delay policy, the completion time policy and the our completion time policy against M and N for a low packet erasure probability. We see that the performance of P-CT and Min-CT are very close. Whereas in Figure 3 and Figure 5 where the sum decoding delay is computed for the same inputs, the performance of P-CT is much better than Min-CT one.

As the channel conditions become harsher (high packet erasure probability), our policy to reduce the completion time minimize the completion time better than the Min-CT. We can see from Figure 2.b, Figure 3.b, Figure 4.ab and Figure 5.b
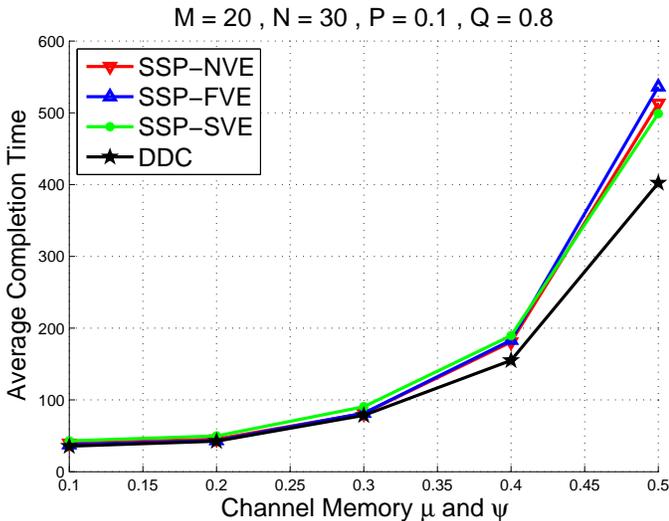
Fig. 11. Average completion time versus chanel memory μ and ψ.

that P-CT outperforms Min-CT in minimizing both the sum decoding delay and the completion time. Figure 6.a shows that for $P > 0.3$, P-CT achieves a significant improvement in the completion time. This can be explained by the light of the P-CT policy characteristics. In the P-CT policy, the number of the erased packets is estimated using the law of large numbers. This approximation can be effective when the erasure of the channel or the input (number of packets and users) are high enough.

From Figure 7, we clearly see that the BPSO algorithm achieves a lower completion time for a low number of iteration (5 iterations). This algorithm have a fixed complexity unlike the multi-layer graph algorithm which have a worst complexity of MN (60 in the figure). This fixed complexity property along with its performance to effectively reduce the completion time make this algorithm more reliable and more suitable to be used.

Figure 10 and Figure 11 shows that our algorithm to control the completion time using the decoding delay in lossy feedback scenario outperforms largely the blind algorithms specially as the channel is more and more persistent (μ increases). The optimist approach (FVE) achieves a reasonable degradation for a low channel persistence. However, this degradation become more severe as the memory of the channel increases. The pessimist approach (NVE) can be seen as the complementary of the optimist approach since it perform better in high memory channel and less in near independent channel. The stochastic approach (SVE) achieves an intermediate result and degrades as the channel is near independent or highly correlated.

IX. CONCLUSION

In this paper, we studied the effect of controlling the decoding delay to reduce the completion time below its currently best known solution for persistent channel. We first derived the decoding-delay-dependent completion time expressions.

We then employed these expressions to design two new heuristic. The first decides on coded packets by reducing the probability of decoding delay increase on a new layering of the IDNC graph based on user criticality in increasing the overall completion time and the second uses binary optimization with multi-layer objective function that preserves prioritization. We, then, extended our study to the limited feedback environment. Simulation results showed that this new algorithm achieves a lower mean completion time and mean decoding delay compared to the best known completion time heuristics, with significant gains in harsh erasure scenarios.

APPENDIX A
AUXILIARY THEOREMS

In this appendix we provide auxiliary theorem that we will use to proof Theorem 4. The following theorem provides the expression of the probability to be in a state of the channel given that the channel was in a particular state at a previous time instant.

Theorem 6. *Let* $(X_n)_{n \geq 1}$ *be a two state (x and y) Markov chain, with* $P_{tr_{x \to y}}$ *and* $P_{tr_{y \to x}}$ *the transition probability from state x to y and y to x, respectively. Let* $\mu = (1 - P_{tr_{x \to y}} - P_{tr_{y \to x}})$ *be the memory of the chain.*

Define $f(n) = \mathbb{P}(X_n = y | X_{n^0} = x), \ \forall \ n \geq n^0$. *We have:*

$$f(n) = P_{tr_{x \to y}} \times \langle \sum_{i=0}^{n-n^0-1} \mu^i \rangle \quad (A.1)$$

where

$$\langle \sum_{x \in X}(.) \rangle = \begin{cases} \sum_{x \in X}(.) & \text{if } X \neq \varnothing \\ 0 & \text{if } X = \varnothing. \end{cases} \quad (A.2)$$

Proof: The proof can be found in [23] Appendix A. ∎

The following theorem gives the expression of the probability to be in a particular state of the channel given the channel realization for the same time instant. Lets consider the channel defined in Figure 1. For notation simplicity, we will not consider the superscripts in this theorem. Let $X_i(n)$ be a random variable that take the value 1 if the transmission at time n is erased and 0 otherwise.

Theorem 7. *The probability of the state of the channel at time n conditioned by the realization $X_i(n)$ at the same time can be expressed as:*

$$\mathbb{P}(\mathcal{C}(n) = y | X_i(n) = x) = \quad (A.3)$$

$$\begin{cases} \dfrac{p\mathcal{P}_G}{p\mathcal{P}_G + q\mathcal{P}_B} & \text{if } x = 1, y = G \\ \dfrac{(1-p)\mathcal{P}_G}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B} & \text{if } x = 0, y = G \\ \dfrac{q\mathcal{P}_B}{p\mathcal{P}_G + q\mathcal{P}_B} & \text{if } x = 1, y = B \\ \dfrac{(1-q)\mathcal{P}_B}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B} & \text{if } x = 0, y = B \end{cases}$$

12

Proof: We first note that using the total probability theorem we have:

$$\mathbb{P}(X_i(n) = x) = \begin{cases} p\mathcal{P}_G + q\mathcal{P}_B & \text{if } x = 1 \\ (1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B & \text{if } x = 0 \end{cases}$$

(A.4)

We now use the Bayes theorem and write:

$$\mathbb{P}(\mathcal{C}(n) = y | X_i(n) = x) =$$
$$\frac{\mathbb{P}(\mathcal{C}(n) = y)}{\mathbb{P}(X_i(n) = x)} \mathbb{P}(X_i(n) = x | \mathcal{C}(n) = y)$$

(A.5)

By simple substitution in the previous expression we obtain:

$$\mathbb{P}(\mathcal{C}(n) = y | X_i(n) = x) =$$

(A.6)

$$\begin{cases} \dfrac{p\mathcal{P}_G}{p\mathcal{P}_G + q\mathcal{P}_B} & \text{if } x = 1, y = G \\ \dfrac{(1-p)\mathcal{P}_G}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B} & \text{if } x = 0, y = G \\ \dfrac{q\mathcal{P}_B}{p\mathcal{P}_G + q\mathcal{P}_B} & \text{if } x = 1, y = B \\ \dfrac{(1-q)\mathcal{P}_B}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B} & \text{if } x = 0, y = B \end{cases}$$

■

APPENDIX B
PROOF OF THEOREM 1

Let us first define $\mathcal{E}_i(t)$ as the cumulative number of transmitted packets from the sender that were erased at user i until time t. It is easy to infer that the reception completion event at time $t = C_i(S)$ of a user i will occur when it receives an instantly decodable packet in the $C_i(S)$-th recovery transmission from the sender. Consequently, $\forall t <= C_i(S)-1$, the transmission at time t following the schedule S can be one of the following options:

- The packet can be erased at user $i \Rightarrow$ The transmission will increase $\mathcal{E}_i(t)$ (i.e. $\mathcal{E}_i(t) = \mathcal{E}_i(t-1) + 1$).
- The packet can be successfully received by the user $\Rightarrow$ Two cases can occur types:
  - The packet is instantly decodable for user i. Note that user i needs to receive $|\mathcal{W}(0)| - 1$ of those packets until time $t = C_i(S) - 1$ in order to complete its reception by the last missing source packet from the transmitted packet at time $t = C_i(S)$. Consequently, the number of such packets received by user i until time $t = C_i(S)$ is equal to $|\mathcal{W}(0)|$.
  - The packet is either non-innovative or non instantly decodable $\Rightarrow$ This will increase the value of $D_i(S)$ by one each time it occurs until the reception completion for this user.

Consequently, the number of recovery transmission sent by the sender following schedule S until user i complete its reception of the frame packets (i.e. completion time of user i) can be expressed as follows:

$$C_i(S) = |\mathcal{W}_i(0)| + D_i(S) + \mathcal{E}_i(C_i(S) - 1).$$

(B.1)

Let $\mathcal{X}_i(t)$ be the number of time instant, from the beginning of the *recovery phase*, until the time t, in which the channel was in Good state and let $\mathcal{Y}_i(t)$ be number in which it was in the bad state. Using the limit distribution of the Markov chain, we can write:

$$\mathcal{X}_i(t) \approx t\mathcal{P}_{G_i^f}$$

(B.2)

$$\mathcal{Y}_i(t) \approx t\mathcal{P}_{B_i^f}$$

(B.3)

$\mathcal{E}_i^g(t)$ and $\mathcal{E}_i^b(t)$ be the number of erased transmission in the Good and Bad state respectively from the beginning of the *recovery phase*, until the time t. Let Using the law of large number in each of the states of the Markov chain, we have:

$$\mathcal{E}_i^g(t) \approx \mathcal{X}_i(t)p_i^f \approx t\mathcal{P}_{G_i^f}p_i^f$$

(B.4)

$$\mathcal{E}_i^b(t) \approx \mathcal{Y}_i(t)q_i^f \approx t\mathcal{P}_{B_i^f}q_i^f$$

(B.5)

For large enough frame size N, the completion time $C_i(S)$ would also be large enough and thus $\mathcal{E}_i(C_i(S) - 1)$ can be approximated using the law of large numbers as follows:

$$\mathcal{E}_i(C_i(S) - 1) = \mathcal{E}_i(t)^g + \mathcal{E}_i(t)^b \approx \alpha_i(C_i(S) - 1),$$

(B.6)

where:

$$\alpha_i = \frac{g_i^f p_i^f + q_i^f b_i^f}{g_i^f + b_i^f}$$

(B.7)

Substituting the previous expression in (B.1) and re-arranging the terms, the completion time for user i can be finally expressed as:

$$C_i(S) \approx \frac{|\mathcal{W}_i(0)| + D_i(S) - \alpha_i}{1 - \alpha_i}.$$

(B.8)

Thus, the expression for the overall completion time can be expressed as:

$$\mathcal{C}(S) \approx \max_{i \in \mathcal{M}} \left\{ \frac{|\mathcal{W}_i(0)| + D_i(S) - \alpha_i}{1 - \alpha_i} \right\}$$

(B.9)

APPENDIX C
PROOF OF THEOREM 3

We first proof that the algorithm, as stated in the original paper, will poorly perform in our system. Let L be the number of particle and T the number of iterations. Assume there is a user i who is missing all the packets (i.e. $\mathcal{H}_i = \varnothing$). Further assume that all users expect of user i received all their packets. Therefore, the only packet combination that can reduce the Wants set of user i, is a packet combination where only one packet is included. We will refer to such packet combination as sparse packet combination. For a random initialisation of one of the particle, the probability that a particle will be sparse is $N\left(\dfrac{1}{2}\right)^N$. Thus, the probability that at least one of the L

particle is spare is:

$$1 - \left(1 - N\left(\frac{1}{2}\right)^N\right)^L \tag{C.1}$$

For a large number of packets N, with high probability none of the initial value of the L particles will be sparse. For a non-sparse particle, the merit function will be 0 since no user will be targeted. As a consequence, the update of the particles will be random since almost all direction will result in a non-sparse particle and thus a 0 merit. Therefore the second iteration of the algorithm can be seen as another initialization of the L particle. The probability to move one particle in a spare configuration after the T iterations of the algorithm is:

$$1 - \left(1 - N\left(\frac{1}{2}\right)^N\right)^{L+T} \tag{C.2}$$

For a small number of iterations T, with high probability, the algorithm will end with a non sparse particles and therefore no update will be made in the system. This process will result in a very poor performance of the overall system. By setting the number of particle equal to the number of packets $L = N$ and using a sparse initialisation of each particle different from the other particles, we can guarantee a decrease in the merit function each time the algorithm is run. Therefore, at each time instant, unless the packet is erased, we can ensure a reduction by at least one packet from the Wants set of user of interest. This conclude to the overall convergence of the system independently of the number of iterations T of the algorithm.

APPENDIX D
PROOF OF THEOREM 4

To compute the probability to loose a transmission $e_i(t)$ or the loose a feedback $f_i(t)$ at time instant t, lets consider the channel defined in Figure 1. For notation simplicity, we will not consider the superscripts in this theorem unless it is necessary to specify the forward and backward channels. We first compute the following probability:

$$\mathbb{P}(X_i(n) = 1 | X_i(n^0) = x), \tag{D.1}$$

where $X_i(n)$ is a random variable that take the value 1 if the transmission at time n is erased and 0 otherwise and $n \geq n_0$. Using the total probability theorem, we write the previous probability as:

$$\begin{aligned}
\mathbb{P}(X_i(n) = 1 | X_i(n^0) = x) = \\
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = G, X_i(n^0) = x) \\
\times \mathbb{P}(\mathcal{C}(n) = G | X_i(n^0) = x) + \\
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = B, X_i(n^0) = x) \\
\times \mathbb{P}(\mathcal{C}(n) = B | X_i(n^0) = x)
\end{aligned} \tag{D.2}$$

By definition of the Markov chain we have:

$$\begin{aligned}
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = G, X_i(n^0) = x) = \\
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = G) = p
\end{aligned} \tag{D.3}$$

$$\begin{aligned}
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = B, X_i(n^0) = x) = \\
\mathbb{P}(X_i(n) = 1 | \mathcal{C}(n) = B) = q
\end{aligned} \tag{D.4}$$

Using the total probability theorem, we can write first term of the previous expression as:

$$\begin{aligned}
\mathbb{P}(\mathcal{C}(n) = G | X_i(n^0) = x) = \\
\mathbb{P}(\mathcal{C}(n) = G | \mathcal{C}(n^0) = G, X_i(n^0) = x) \\
\times \mathbb{P}(\mathcal{C}(n^0) = G | X_i(n^0) = x) + \\
\mathbb{P}(\mathcal{C}(n) = G | \mathcal{C}(n^0) = B, X_i(n^0) = x) \\
\times \mathbb{P}(\mathcal{C}(n^0) = B | X_i(n^0) = x)
\end{aligned} \tag{D.5}$$

We can further reduce the previous expressions:

$$\begin{aligned}
\mathbb{P}(\mathcal{C}(n) = G | X_i(n^0) = x) = \\
\mathbb{P}(\mathcal{C}(n) = G | \mathcal{C}(n^0) = G) \mathbb{P}(\mathcal{C}(n^0) = G | X_i(n^0) = x) + \\
\mathbb{P}(\mathcal{C}(n) = G | \mathcal{C}(n^0) = B) \mathbb{P}(\mathcal{C}(n^0) = B | X_i(n^0) = x)
\end{aligned} \tag{D.6}$$

Similarly, we apply the same development to the second term:

$$\begin{aligned}
\mathbb{P}(\mathcal{C}(n) = B | X_i(n^0) = x) = \\
\mathbb{P}(\mathcal{C}(n) = B | \mathcal{C}(n^0) = G) \mathbb{P}(\mathcal{C}(n^0) = G | X_i(n^0) = x) + \\
\mathbb{P}(\mathcal{C}(n) = B | \mathcal{C}(n^0) = B) \mathbb{P}(\mathcal{C}(n^0) = B | X_i(n^0) = x)
\end{aligned} \tag{D.7}$$

Using Theorem 6 and Theorem 7, we can express the previous probability according to the value of x:

$$\mathbb{P}(X_i(n) = 1 | X_i(n^0) = x) = \tag{D.8}$$

$$\begin{cases}
\dfrac{p\mathcal{P}_G}{p\mathcal{P}_G + q\mathcal{P}_B}(p + (q-p)b \sum\limits_{i=0}^{n-n^0-1} \mu^i) \\
\quad + \dfrac{q\mathcal{P}_B}{p\mathcal{P}_G + q\mathcal{P}_B}(q + (p-q)g \sum\limits_{i=0}^{n-n^0-1} \mu^i) \\
\quad \text{if } x = 1 \\[4pt]
\dfrac{(1-p)\mathcal{P}_G}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B}(p + (q-p)b \sum\limits_{i=0}^{n-n^0-1} \mu^i) \\
\quad + \dfrac{(1-q)\mathcal{P}_B}{(1-p)\mathcal{P}_G + (1-q)\mathcal{P}_B}(q + (p-q)g \sum\limits_{i=0}^{n-n^0-1} \mu^i) \\
\quad \text{if } x = 0
\end{cases}$$

We now apply our framework to compute the probability to loose a transmission and to loose the feedback. At time t_i^0, packet j^0 was attempted to user i. Therefore, the probability to loose the transmission can be seen as:

$$e_i(t) = \mathbb{P}(X_i(n) = 1 | X_i(t_i^0) = x) = $$
$$\begin{cases}
\mathbb{P}(X_i(n) = 1 | X_i(t_i^0) = 1) \text{ if } f_{ij^0} = 1 \\
\mathbb{P}(X_i(n) = 1 | X_i(t_i^0) = 0) \text{ if } f_{ij^0} = 0
\end{cases} \tag{D.9}$$

Using the expression derived above, we can write this

probability as:

$$e_i = \tag{D.10}$$

$$
\begin{cases}
\dfrac{p_i^f g_i^f}{p_i^f g_i^f + q_i^f b_i^f}(p_i^f + (q_i^f - p_i^f)b_i^f \displaystyle\sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\
\quad + \dfrac{q_i^f b_i^f}{p_i^f g_i^f + q_i^f b_i^f}(q_i^f + (p_i^f - q_i^f)g \displaystyle\sum_{i=0}^{t-n^0-1} \mu^i) \\
\qquad \text{if } f_{ij^0} = 1 \\[4pt]
\dfrac{(1-p_i^f)g_i^f}{(1-p_i^f)g_i^f + (1-q_i^f)b_i^f} \\
\qquad\quad \times (p_i^f + (q_i^f - p_i^f)b_i^f \displaystyle\sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\
\quad + \dfrac{(1-q_i^f)b_i^f}{(1-p_i^f)g_i^f + (1-q_i^f)b_i^f} \\
\qquad\quad \times (q_i^f + (p_i^f - q_i^f)g_i^f \displaystyle\sum_{i=0}^{t-t_i^{(0)}-1} \mu^i) \\
\qquad \text{if } f_{ij^0} = 0
\end{cases}
$$

At time t_i^* the feedback was successfully received from user i. Thus, the probabilities $f_i(t)$ of loosing a feedback from user i at time $t > t_i^*$ can be expressed as:

$$f_i = \mathbb{P}(X_i(n) = 1 | X_i(t_i^*) = 0)$$

$$= \frac{(1-p_i^b)g_i^b}{(1-p_i^b)g_i^b + (1-q_i^b)b_i^b}(p_i^b + (q_i^b - p_i^b)b_i^b \sum_{i=0}^{t-t_i^*-1} \psi^i)$$

$$+ \frac{(1-q_i^b)b_i^b}{(1-p_i^b)g_i^b + (1-q_i^b)b_i^b}(q_i^b + (p_i^b - q_i^b)g_i^b \sum_{i=0}^{t-t_i^*-1} \psi^i) \tag{D.11}$$

Note that we can obtain the expressions derived in [23] by setting $p = 0$ and $q = 1$:

$$e_i = \begin{cases} 1 - g_i^f \displaystyle\sum_{i=0}^{n-t_i^0-1} \mu^i & \text{if } f_{ij^0} = 1 \\ b_i^f \displaystyle\sum_{i=0}^{n-t_i^0-1} \mu^i & \text{if } f_{ij^0} = 0 \end{cases} \tag{D.12}$$

$$f_i = b_i^b \sum_{i=0}^{n-t_i^*-1} \psi^i \tag{D.13}$$

APPENDIX E
EXPRESSIONS OF INNOVATIVE AND FINISH PROBABILITIES

Let $\mathcal{K}_{ij}$ be the set of indexes of the frames in which packet j was attempted to user i since the last time the BS received feedback from this user, excluding the current frame. Define $\mathcal{U}_i^d(n)$ as the following:

$$\mathcal{U}_i^d(n) = \bigcup_{j \in \mathcal{W}_i(n \times T_f)} \lambda_{ij}(n), \; \forall \, n \in \mathbb{N}^+. \tag{E.1}$$

Given these definitions, the probability $p_{i,n}(j,t)$ that packet

j is innovative for user i can be expressed as:

$$p_{i,n}(j,t) = \langle \prod_{k \in \lambda_{ij}(n^+(t))} e_i(k) \rangle$$

$$\times \langle \prod_{k \in \mathcal{K}_{ij}} \rangle \left\{ \left(\prod_{s \in \mathcal{U}_i^d(k)} e_i(s) + \prod_{s \in \lambda_{ij}(k)} e_i(s) \right. \right.$$

$$\times (1 - \prod_{s \in \mathcal{U}_i^d(k) \setminus \lambda_{ij}(k)} e_i(s)) f_i(u_i(k)) \Bigg)$$

$$\times \left(\prod_{s \in \mathcal{U}_i^d(k)} e_i(s) + (1 - \prod_{s \in \mathcal{U}_i^d(k)} e_i(s)) f_i(u_i(k)) \right)^{-1} \Bigg\}$$
$$\tag{E.2}$$

The probability $p_{i,f}(t)$ that user i successfully received all his primary packets but $\mathcal{W}_i(t) \neq \varnothing$, at time t is the following:

$$p_{i,f}(t) = \prod_{j \in \mathcal{W}_i(t)} (1 - p_{i,n}(j,t)), \tag{E.3}$$

where $u_i(n) = n * T_f - T_u + T_{u_i}$ and

$$\langle \prod_{x \in X} (.) \rangle = \begin{cases} \displaystyle\prod_{x \in X} (.) & \text{if } X \neq \varnothing \\ 1 & \text{if } X = \varnothing. \end{cases} \tag{E.4}$$

Proof: The proof can be found in [23] Appendix D. $\blacksquare$

APPENDIX F
PROOF OF THEOREM 5

Let $\mathbb{M}(t)$ be the event that $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\} > \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\}$ at time after a transmission $\kappa(t)$ at time t. The probability of this event can be expressed as:

$$\mathbb{P}(\mathbb{M}(t)) = \mathbb{P}\left(\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\} > \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\} \right)$$

$$= 1 - \mathbb{P}\left(\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\} = \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\} \right). \tag{F.1}$$

Users $j \in \mathcal{M} \setminus \mathcal{P}(t)$ are unable to increase $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\}$ compared to $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\}$ with probability 1, even if they experience a decoding delay. This is true since the set $\mathcal{P}(t)$ is constructed such that it contains all users that have non-zero probabilities of increasing the completion time. According the definition of $\mathcal{C}_i(t)$ in (10), $\forall \, i \in \mathcal{M}$, users $i \in \mathcal{P}(t)$ will not increase $\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\}$ after the transmission $\kappa(t)$ only if they do not experience a decoding delay increment in this transmission. Consequently, we get:

$$\mathbb{P}\left(\max_{i \in \mathcal{M}} \{\mathcal{C}_i(t)\} = \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\} \right)$$

$$= \mathbb{P}\left(\max_{i \in \mathcal{P}(t)} \{\mathcal{C}_i(t)\} = \max_{i \in \mathcal{M}} \{\mathcal{C}_i(t-1)\} \right)$$

$$= \mathbb{P}(D_i(t) - D_i(t-1) = 0, \forall \, i \in \mathcal{P}(t))$$

$$= \prod_{i \in \mathcal{P}(t)} \mathbb{P}(D_i(t) - D_i(t-1) = 0). \tag{F.2}$$

According to the analysis done in [23], the probability of the decoding delay increase for user i is given by the following theorem:

Theorem 8. *The probability that user i does not experience a decoding delay at time t, after the transmission κ is:*

$$\mathbb{P}(d_i(\kappa, t) = 0)$$
$$= \begin{cases} e_i(t) & i \in (\widehat{\tau} \cap \overline{F}) \\ e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t) & i \in (\widehat{\tau} \cap F) \\ 1 & i \in (\tau \cap \overline{U}) \\ e_i(t) + p_{i,n}(\kappa_i, t) & i \in (\tau \cap (U \setminus F)) \\ \quad - e_i(t)p_{i,n}(\kappa_i, t) \\ e_i(t) + (1 - e_i(t))(p_{i,n}(\kappa_i, t) & i \in (\tau \cap F) \\ \quad + p_{i,f}(t)), \end{cases}$$
$$\text{(F.3)}$$

where $\widehat{\tau}$ is set of users not targeted and having non-empty Wants sets (i.e. $\widehat{\tau} = M_w \setminus \tau$), κ_i is the intended packet for user i in the transmission κ, F is the set of users having all their remaining packets in an uncertain state and U is the set of users having the intended packet for them in an uncertain state. The notation $\overline{X}$ refers to the set complementary to the set X.

Proof: The proof can be found in [23] Appendix A. ∎

In other words, the completion time does not increase only if all the users having the completion time so far do not experience a decoding delay increase in the next transmission. Using the expression of the decoding delay increase, the probability of event $\mathbb{M}(t)$ to occur can be expressed as follows:

$$\mathbb{P}(\mathbb{M}(t)) = 1 - \prod_{i \in \mathcal{P}(t)} \mathbb{P}(d_i(\kappa, t) = 0)$$
$$= 1 - \prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} e_i(t)$$
$$\prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap F)} e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t)$$
$$\prod_{i \in (\mathcal{P}(t) \cap \tau \cap (U \setminus F))} e_i(t) + p_{i,n}(\kappa_i, t) - e_i(t)p_{i,n}(\kappa_i, t)$$
$$\prod_{i \in (\mathcal{P}(t) \cap \tau \cap F)} e_i(t) + (1 - e_i(t))(p_{i,n}(\kappa_i, t) + p_{i,f}(t)).$$
$$\text{(F.4)}$$

From the expressions of the completion time increment, we can express the minimum completion time problem as a maximum weight clique problem, such that:

$$\kappa^*(t) = \underset{\kappa(t) \in \mathcal{G}}{\operatorname{argmin}} \{\mathbb{P}(\mathbb{M}(t))\}$$
$$= \underset{\kappa(t) \in \mathcal{G}}{\operatorname{argmin}} \left\{ 1 - \prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} e_i(t) \right.$$
$$\prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap F)} e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t)$$
$$\prod_{i \in (\mathcal{P}(t) \cap \tau \cap (U \setminus F))} e_i(t) + p_{i,n}(\kappa_i, t) - e_i(t)p_{i,n}(\kappa_i, t)$$
$$\left. \prod_{i \in (\mathcal{P}(t) \cap \tau \cap F)} e_i(t) + (1 - e_i(t))(p_{i,n}(\kappa_i, t) + p_{i,f}(t)) \right\}$$
$$= \underset{\kappa(t) \in \mathcal{G}}{\operatorname{argmax}} \left\{ \prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} e_i(t) \right.$$
$$\prod_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap F)} e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t)$$
$$\prod_{i \in (\mathcal{P}(t) \cap \tau \cap (U \setminus F))} e_i(t) + p_{i,n}(\kappa_i, t) - e_i(t)p_{i,n}(\kappa_i, t)$$
$$\left. \prod_{i \in (\mathcal{P}(t) \cap \tau \cap F)} e_i(t) + (1 - e_i(t))(p_{i,n}(\kappa_i, t) + p_{i,f}(t)) \right\}.$$
$$\text{(F.5)}$$

Since the function $log(.)$ is an increasing function, then the problem can be expressed as:

$$\kappa^*(t) = \underset{\kappa(t) \in \mathcal{G}}{\operatorname{argmin}} \log \{\mathbb{P}(\mathbb{M}(t))\}$$
$$= \underset{\kappa(t) \in \mathcal{G}}{\operatorname{argmax}} \left\{ \sum_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} \log(e_i(t)) \right.$$
$$+ \sum_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap F)} \log(e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t))$$
$$+ \sum_{i \in (\mathcal{P}(t) \cap \tau \cap (U \setminus F))} \log(e_i(t) + p_{i,n}(\kappa_i, t) - e_i(t)p_{i,n}(\kappa_i, t))$$
$$\left. + \sum_{i \in (\mathcal{P}(t) \cap \tau \cap F)} \log(e_i(t) + (1 - e_i(t))(p_{i,n}(\kappa_i, t) + p_{i,f}(t))) \right\}.$$
$$\text{(F.6)}$$

If user i does not have all its wanted packets in an uncertain state, then the probability than he finished receiving all its wanted packet is 0. Thus $p_{i,f}(t) = 0$, $\forall\, i \in \overline{F}$. Therefore, we have:

$$\sum_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} \log(e_i(t))$$
$$= \sum_{i \in (\mathcal{P}(t) \cap \widehat{\tau} \cap \overline{F})} \log(e_i(t) + p_{i,f}(t) - e_i(t)p_{i,f}(t)). \quad \text{(F.7)}$$

Using (F.7), the expression below can be simplified as:

$$\sum_{i\in(\mathcal{P}(t)\cap\hat{\tau}\cap\overline{F})} \log(e_i(t))$$

$$+ \sum_{i\in(\mathcal{P}(t)\cap\hat{\tau}\cap F)} \log(e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t))$$

$$= \sum_{i\in(\mathcal{P}(t)\cap\hat{\tau})} \log(e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)). \quad \text{(F.8)}$$

It is clear that $(U\setminus F)\subseteq\overline{F}$, then $p_{i,f}(t)=0$, $\forall\, i\in(U\setminus F)$. We then obtain:

$$\sum_{i\in(\mathcal{P}(t)\cap\tau\cap(U\setminus F))} \log(e_i(t)+p_{i,n}(\kappa_i,t)-e_i(t)p_{i,n}(\kappa_i,t))$$

$$= \sum_{i\in(\mathcal{P}(t)\cap\tau\cap(U\setminus F))} \log(e_i(t)$$

$$+ (1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))). \quad \text{(F.9)}$$

Therefore, using (F.9), we can simplify the below expression:

$$\sum_{i\in(\mathcal{P}(t)\cap\tau\cap(U\setminus F))} \log(e_i(t)+p_{i,n}(\kappa_i,t)-e_i(t)p_{i,n}(\kappa_i,t))$$

$$+ \sum_{i\in(\mathcal{P}(t)\cap\tau\cap F)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t)))$$

$$= \sum_{i\in(\mathcal{P}(t)\cap\tau\cap U)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))). \quad \text{(F.10)}$$

Given the above simplifications, the maximum weight clique problem can be written as follows:

$$\kappa^*(t) = \operatorname*{argmin}_{\kappa(t)\in\mathcal{G}} \log\{\mathbb{P}(\mathbb{M}(t))\}$$

$$= \operatorname*{argmax}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\hat{\tau})} \log(e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)) \right.$$

$$+ \sum_{i\in(\mathcal{P}(t)\cap\tau\cap U)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))) \Bigg\}$$

$$= \operatorname*{argmin}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log(e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)) \right.$$

$$- \sum_{i\in(\mathcal{P}(t)\cap\tau\cap U)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))) \Bigg\}. \quad \text{(F.11)}$$

Note that if the targeted packet κ_i of user i is not an uncertain packet (i.e. $i\in\overline{U}$), then the packet is certainly innovative. Since that this user have at least one certain wanted packet then he surely still needs packets. In other words, we have $i\in\overline{U}\Rightarrow p_{i,n}(\kappa_i,t)=1$ and $p_{i,f}(t)=0$. We write the following expression as:

$$\sum_{i\in(\mathcal{P}(t)\cap\tau)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t)))$$

$$= \sum_{i\in(\mathcal{P}(t)\cap\tau\cap U)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t)))$$

$$+ \sum_{i\in(\mathcal{P}(t)\cap\tau\cap\overline{U})} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t)))$$

$$= \sum_{i\in(\mathcal{P}(t)\cap\tau\cap U)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))). \quad \text{(F.12)}$$

Giving all the above simplifications, we now express the maximum weight clique problem as:

$$\kappa^*(t) = \operatorname*{argmin}_{\kappa(t)\in\mathcal{G}} \log\{\mathbb{P}(\mathbb{M}(t))\}$$

$$= \operatorname*{argmin}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log(e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)) \right.$$

$$- \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log(e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))) \Bigg\}$$

$$= \operatorname*{argmin}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log\left(\frac{e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)}{e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))}\right) \right\}$$

$$= \operatorname*{argmax}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log\left(\frac{e_i(t)+(1-e_i(t))(p_{i,n}(\kappa_i,t)+p_{i,f}(t))}{e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)}\right) \right\}$$

$$= \operatorname*{argmax}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log\left(1+\frac{(1-e_i(t))p_{i,n}(\kappa_i,t)}{e_i(t)+p_{i,f}(t)-e_i(t)p_{i,f}(t)}\right) \right\}$$

$$= \operatorname*{argmax}_{\kappa(t)\in\mathcal{G}} \left\{ \sum_{i\in(\mathcal{P}(t)\cap\tau)} \log\left(1+\frac{p_{i,n}(\kappa_i,t)}{\frac{e_i(t)}{1-e_i(t)}+p_{i,f}(t)}\right) \right\}. \quad \text{(F.13)}$$

In other words, the transmission $\kappa(t)$ than can satisfy the critical criterion can be selected using a maximum weight clique problem in which the weight of each vertex v_{ij} in $\mathcal{P}(t)$

can be expressed as:

$$w_{ij}^* = \log \left(1 + \frac{p_{i,n}(j,t)}{\dfrac{e_i(t)}{1 - e_i(t)} + p_{i,f}(t)} \right).$$ (F.14)

References

[1] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Completion time reduction in instantly decodable network coding through decoding delay control," in *Proc. of IEEE Global Telecommunications Conference, (GLOBECOM' 2014), Austin, Texas, USA, available ArXiv e-prints*, Dec. 2014.

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[3] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. of IEEE Global Telecommunications Conference, (GLOBECOM' 2010), Miami, Florida, USA*, Dec. 2010, pp. 1–5.

[4] M. Muhammad, M. Berioli, G. Liva, and G. Giambene, "Instantly decodable network coding protocols with unequal error protection," in *Proc. of IEEE International Conference on Communications (ICC' 2013),Sydney, Australia*, June 2013, pp. 5120–5125.

[5] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, *Accepted in IEEE Wireless Communications Letters*.

[6] S. Sorour, N. Aboutorab, P. Sadeghi, M. S. Karim, T. Al-Naffouri, and M.-S. Alouini, "Delay reduction in persistent erasure channels for generalized instantly decodable network coding," *Proc. of IEEE Vehicular Technology Conference, (VTC' 2013), Dresden, Germany*, pp. 1–5, June. 2013.

[7] S. Tajbakhsh, P. Sadeghi, and N. Aboutorab, "Instantly decodable network codes for cooperative index coding problem over general topologies," in *Australian Communications Theory Workshop (AusCTW' 2014)*, Feb 2014, pp. 84–89.

[8] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. of IEEE International Conference on Communications, (ICC' 2010), Cape Town, South Africa*, May, 2010, pp. 1–5.

[9] L. Lu, M. Xiao, and L. Rasmussen, "Design and analysis of relay-aided broadcast using binary network codes," *Journal of Communications (JCM), Special Issue on Advances in Communications and Networking, Nov. 2011*.

[10] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Workshop on Network Coding, Theory, and Applications, (NetCod' 2009), Lausanne, Switzerland*, June 2009.

[11] Y. Liu and C. W. Sung, "Quality-aware instantly decodable network coding," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1604–1615, March 2014.

[12] 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*, vol. 2010, no. 1, 2010.

[13] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding (idnc) schemes for two unicast sessions with hard deadline constraints," in *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton' 2011),Monticello, IL, USA*, Sept 2011, pp. 784–791.

[14] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. of IEEE International Symposium on Information Theory, (ISIT' 2009), Seoul, Korea*, July 2009, pp. 844–848.

[15] L. Lu, M. Xiao, and L. Rasmussen, "Relay-aided broadcasting with instantaneously decodable binary network codes," in *Proc. of IEEE International Conference on Computer Communication Networks, (ICCCN), Maui, Hawaii, August, 2011*.

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

[17] M. Esmaeilzadeh and P. Sadeghi, "Optimizing completion delay in network coded systems over tdd erasure channels with memory," in *Proc. of IEEE International Symposium on Communications and Information Technologies, (ISCIT' 2012), Queensland, Australia*, Oct. 2012, pp. 883–888.

[18] S. Sorour and S. Valaee, "On densifying coding opportunities in instantly decodable network coding graphs," in *Proc. of IEEE International Symposium on Information Theory Proceedings, (ISIT' 2012), Cambridge, MA, USA*, July 2012, pp. 2456–2460.

[19] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *ArXiv e-prints*, Jan. 2012.

[20] S. Sorour, A. Douik, S. Valaee, T. Y. Al-Naffouri, and M.-S. Alouini, "Partially blind instantly decodable network codes for lossy feedback environment," *Accepted in IEEE Transactions on Wireless Communications, available ArXiv e-prints*.

[21] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE Transactions on Selected Areas in Communications*, vol. 29, no. 5, pp. 1094–1105, 2011.

[22] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Delay reduction in lossy intermittent feedback for generalized instantly decodable network coding," in *Proc. of IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications, (WiMob' 2013), Lyon, France*, Oct. 2013.

[23] ——, "Delay minimization for instant decodable network coding in persistent channels with feedback intermittence," *ArXiv e-prints*, Jul. 2013.

[24] N. Aboutorab, P. Sadeghi, and S. Sorour, "On Improving the Balance between the Completion Time and Decoding Delay in Instantly Decodable Network Coded Systems," *ArXiv e-prints*, Nov. 2013.

[25] H. Tracey and S. Desmond, *Network Coding: An Introduction*. Cambridge, April, 2008.

[26] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," *ArXiv e-prints*, Mar. 2013.

[27] P. Sadeghi, R. 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, 2008.

[28] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963. [Online]. Available: http://dx.doi.org/10.1002/j.1538-7305.1963.tb00955.x

[29] M. Mushkin and I. Bar-David, "Capacity and coding for the gilbert-elliott channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277–1290, 1989.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of IEEE International Conference on Neural Networks, (ICNN' 95), Perth, Western Australia*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[31] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. of the 6th International Symposium on Micro Machine and Human Science, (MHS '95), Nagoya, Japan*, Oct 1995, pp. 39–43.

[32] J. Kennedy and R. Eberhart, *Swarm Intelligence*. Morgan Kaufmann Publishers, Inc., 2001.

[33] ——, "A discrete binary version of the particle swarm algorithm," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA.*, vol. 5, Oct 1997, pp. 4104–4108 vol.5.

[34] M. Khanesar, M. Teshnehlab, and M. Shoorehdeli, "A novel binary particle swarm optimization," in *Mediterranean Conference on Control Automation, (MED' 07), Athens, Greece.*, June 2007, pp. 1–6.

[35] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "On minimizing the maximum broadcast decoding delay for instantly decodable network coding," *Submitted to proc of IEEE Vehicular Technology Conference, (VTC'Fall 2014), available ArXiv e-prints*, Apr. 2014.

[36] S. Sorour and S. Valaee, "Completion delay reduction in lossy feedback scenarios for instantly decodable network coding," in *Proc. of IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications, (PIMRC' 2011), Toronto, Canada*, Sept. 2011, pp. 2025–2029.