

# Delay Reduction in Multi-Hop Device-to-Device Communication using Network Coding

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

**Abstract**—This paper considers the problem of reducing the broadcast decoding delay of wireless networks using instantly decodable network coding (IDNC) based device-to-device (D2D) communications. In a D2D configuration, devices in the network can help hasten the recovery of the lost packets of other devices in their transmission range by sending network coded packets. Unlike previous works that assumed fully connected network, this paper proposes a partially connected configuration in which the decision should be made not only on the packet combinations but also on the set of transmitting devices. First, the different events occurring at each device are identified so as to derive an expression for the probability distribution of the decoding delay. The joint optimization problem over the set of transmitting devices and the packet combinations of each is, then, formulated. The optimal solution of the joint optimization problem is derived using a graph theory approach by introducing the cooperation graph and reformulating the problem as a maximum weight clique problem in which the weight of each vertex is the contribution of the device identified by the vertex. Through extensive simulations, the decoding delay experienced by all devices in the Point to Multi-Point (PMP) configuration, the fully connected D2D (FC-D2D) configuration and the more practical partially connected D2D (PC-D2D) configuration are compared. Numerical results suggest that the PC-D2D outperforms the FC-D2D and provides appreciable gain especially for poorly connected networks.

**Index Terms**—Instantly decodable network coding, device-to-device communications, delay reduction, partially connected network, maximum weight clique problem.

## I. INTRODUCTION

### A. Background

*Network Coding (NC)* [2] is a promising technique to significantly improve the throughput (i.e., the network capacity) and to minimize delay over wireless erasure channels. These benefits are of great interest for the proliferation and spread of real-time applications which require quick and reliable packet transmission over lossy channels with low latency [3], such as streaming, cellular, satellite networks, and Internet television.

A part of this paper is accepted in IEEE International Symposium on Network Coding (NetCod' 2015), Sydney, Australia. The pre-print version is available at [1].

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 Electrical Engineering Department at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Eastern Province, Saudi Arabia, e-mail: tareq.alnaffouri@kaust.edu.sa.

Hong-Chuan Yang is with the Department of Electrical and Computer Engineering, University of Victoria (UVIC), British Columbia, Canada, email: hy@uvic.ca.

Two classes of NC can be distinguished in the literature, namely the Random Network Coding (RNC) [4], [5] and the Opportunistic Network Coding (ONC) [6], [7]. While the sender in RNC combines packets using random coefficients, ONC exploits the diversity of received and lost packets to generate the network encoded packet combinations. Even though RNC is optimal in reducing the number of transmissions even without feedback, it is not suitable for real-time applications since it does not enable progressive decoding of the frame. Moreover, unlike ONC, the computation complexity of RNC is prohibitive for real-time applications.

This paper is interested in delay sensitive broadcast applications in which each device should receive all the packets within a frame with minimum delay [8]. A suitable technique for such applications is the Instantly Decodable Network Coding (IDNC) [9]. In this subclass of ONC, the sender encodes the packets using binary XOR and devices decode them by the same means, which is an important property that ensures fast encoding/decoding and overcomes the computationally expensive matrix inversion operations. In addition, non-instantly decodable packets are not stored, which eliminates the need for buffers and allows the design of cost-efficient receivers. Thanks to its numerous benefits, IDNC was an subject of intensive research in the past few years [8]–[18].

In the wireless medium, packet losses, occurring due to many phenomena (e.g., the mobility and the propagation environment), can be seen as packet erasures at higher communication layers [19]. The erasure nature of the links affects the ability of devices to decode the information flow synchronously and thus affects the delivery of meaningful data. As a consequence, a better use of the channel does not usually translates to a better throughput at higher communication layers [19] which motivates the definition of the various delay metric in NC. The commonly used delay metrics in IDNC are the completion time and the decoding delay. The former definition considers the delay as the overall transmission time and the latter as the individual delay when a transmitted packet brings no new information at its reception instant. This paper considers the minimization of the decoding delay as it represents a crucial step to study the completion time reduction [20].

### B. Related Work

Determining the packet combinations for the whole recovery phase to optimally reduce the decoding delay is shown to be intractable even for erasure free [21] or the off-line [22] scenarios (i.e., scenarios in which the erasure events' are known in advance) with only three devices. In order to overcome the complexity, to the best of the authors' knowledge, the

most adopted approach [9], [12]–[18] is to study the on-line decoding delay. Therefore, the decoding delay minimization problem is formulated for each recovery transmission.

In all aforementioned works, the wireless centralized sender of a Point to Multi-Point (PMP) network (such as cellular, Wi-Fi and roadside to vehicle systems) is the only transmitter in charge of both the packet sending and recovery processes. This approach consumes a lot of the sender resources and threatens its ability to deliver the packets with the desired rates. The problem is expected to further escalate in the next generation mobile radio system (5G), since the required data rates and the Quality of Service (QoS) requirements are becoming even more constraining [23], [24]. The notion of Device-to-device (D2D) communications, introduced in [25], is a suitable technique to overcome the problem. In a D2D configuration, devices can take care of the packet recovery process by exchanging packets with each other over short range and possibly more reliable channels. This D2D model also provides fast and secure data communications over ad-hoc networks, e.g., wireless sensor networks. Minimizing delays in such IDNC-based D2D systems is of great interest.

Unlike the PMP scenario in which the decision is made only on the packet combination to be transmitted, the decision in a D2D environment should be made also on the set of transmitting devices in every transmission to achieve the best network performance. Aboutorab et al. [26] investigate the problem of minimizing the sum decoding delay in a centralized D2D environment, where the term centralized refers to the configuration in which a *leader* takes all the decisions. The authors in [27] extend the study to the imperfect feedback D2D networks. In a fully distributed D2D system, reference [28] considers the completion time minimization using game theory as a tool to improve the distributed solution. The formulation is further extended to the decoding delay reduction in [29].

These prior works on IDNC-based D2D systems assumes that the network is fully connected (FC-D2D), i.e., each device can target all other devices over one-hop transmission, and thus only one device can transmit at each transmission slot. This assumption of FC-D2D may not apply in some realistic scenarios due to the short device transmission ranges and their widespread over the large cell area. Longer-range D2D transmissions can easily limit the desirable property of more reliable communications between devices compared to those from the centralized sender. This work proposes to study of the decoding delay minimization in the case of partially connected D2D networks (PC-D2D). The partially connected configuration, in which the fully connected can be seen as a particular case, add a new dimension to the problem as many devices can communicate simultaneously each with different packet combination.

### C. Contribution

The main contribution of this paper is to study the reduction of the decoding delay of IDNC in partially connected D2D networks. Whereas in fully connected D2D networks the problem of selecting the transmitting devices and the problem of selecting the packet combinations to be transmitted can be

addressed separately, in partially connected systems the two problems are interdependent. Therefore, the optimization over the set of transmitting devices and their packet combinations should be addressed jointly. Although the paper focuses on a centralized solution, the decoding delay analysis presented in this paper will serve as a reference to future research direction on fully distributed systems since it provides a lower bound on the achievable decoding delay. Moreover, the decoding delay formulation is useful to study the completion time reduction problem [20].

The paper first identify the different expected decoding delay for each device in the network. These expressions are then exploited to formulate the minimum decoding delay problem in IDNC-based PC-D2D networks. To solve the problem, the paper first proposes a decoupling approach. The variables of the problem are shown to be separable in the interference-less scenario. The paper addresses the interference-less decoding delay reduction using a graph theory approach by introducing the local IDNC graph for packet generation and the cooperation graph and reformulating the problem as a maximum weight clique problem that can be efficiently solved using existing literature from graph theory [30], [31]. The solution in the interference-less scenario is further combined with a clustering technique to solve the original problem. Finally, the paper proposes a method to generate the minimum number of clusters to reach the optimal solution of the decoding delay reduction problem. Simulations results show that the proposed solution displays appreciable gain as compared with the fully connected D2D network approach and the PMP configuration.

The rest of this paper is organized as follows: Section II introduces the system model and parameters. The decoding delay minimization problem in partially connected D2D network is formulated in Section III. Section IV solves the problem in the particular interference-less case. In Section V the decoding delay minimization problem is solved. Simulation results are illustrated and discussed in Section VI before concluding in Section VII.

## II. SYSTEM MODEL AND NOTATIONS

### A. System Model and Parameters

Consider a wireless base-station (BS) that desires to transmit  $N$  different packets to a set  $\mathcal{M}$  of  $M$  device. Let  $\mathcal{N}$  denote the frame consisting of the various source packets. The BS begins by broadcasting the  $N$  source packets sequentially. Each device listens and sends an acknowledgement upon each successful reception. The probability of a packet loss at device  $i$  when the BS is transmitting is  $q_i$ ,  $\forall i \in \mathcal{M}$ . At the end of this initial phase, we assume that each packet of the frame  $\mathcal{N}$  is acknowledged by at least one device. Otherwise, the BS keeps broadcasting the packet until the condition is verified. After the initialization phase, the packets of the frame can be in one of the following two sets for each device  $i$ :

- The Has set  $\mathcal{H}_i$ : the set of packets received by device  $i$ .
- The Wants set  $\mathcal{W}_i$ : the set of packets lost by device  $i$ .

In the recovery phase, devices cooperate to ensure that everyone successfully receives all the  $N$  packets. The transmitting devices and the packet combinations are chosen using the

available information including the expected erasure patterns of the network links, and the diversity of received/lost packets across the network. In this phase, the coded packets can be one of the following options for device  $i$ :

- *Non-innovative*: A packet is non-innovative for device  $i$  if it does not contain *any* packet from  $\mathcal{W}_i$ .
- *Instantly Decodable*: A packet is instantly decodable for device  $i$  if it contains exactly *one* packet from  $\mathcal{W}_i$ .
- *Non-Instantly Decodable*: A packet is non-instantly decodable for device  $i$  if it contains more than one packet from  $\mathcal{W}_i$ .

The decoding delay [12], [28] is defined as follows:

**Definition 1.** *At any recovery phase transmission, a device  $i$ , with non-empty Wants sets, experiences one unit increase of decoding delay if he cannot hear exactly one transmission or if he hears a packet that is either non-instantly decodable or non-innovative.*

Let  $\mathbf{C} = [c_{ij}]$ ,  $\forall (i, j) \in \mathcal{M}^2$  be the connectivity matrix representing the connectivity between any pair of devices defined as follows:

$$c_{ij} = \begin{cases} 1 & \text{if devices } i, j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}, \forall (i, j) \in \mathcal{M}^2.$$

Note that  $\mathbf{C}$  is a symmetric matrix that depends on the network topology (i.e., the relative positions of devices in the network). This paper considers a network with a general topology. However, each device should be able to target any other device through single or multi-hop transmission (via the intermediate nodes). In graph theory terms, the graph representing the devices is *connected* or equivalently the matrix  $\mathbf{C}$  is *connected*.

Let  $\mathcal{C}_i$  be the coverage zone of device  $i$  defined as the set of devices in the transmission range of device  $i$  (i.e.,  $\mathcal{C}_i = \{j \in \mathcal{M} \mid c_{ij} = 1\}$ ), and let  $\mathbf{P} = [p_{ij}]$ ,  $\forall (i, j) \in \mathcal{M}^2$  denote the packet erasure probability from device  $i$  to device  $j$ .

## B. Notations

The notation  $\overline{X}$ , where  $X$  is a set in the ensemble  $E$ , refers to the complementary of the set, i.e.,  $\overline{X} = E \setminus X$ . The notation  $\mathcal{P}(E)$  refers to the power set of the ensemble  $E$ . In other words,  $\mathcal{P}(E)$  is the set of all the subsets of the ensemble  $E$ . A partition  $\{X_i\}_{1 \leq i \leq n}$  of the ensemble  $E$  is denoted by  $E = \bigoplus_{i=1}^n X_i$ . Let  $|X|$  denote the cardinality of the set  $X$ .

## III. MINIMUM DECODING DELAY PROBLEM FORMULATION

This section first formulates the minimum decoding delay problem in partially connected D2D networks. The problem is expressed as a joint optimization over the set of transmitting devices and the packet combinations to be transmitted. Further, the section illustrates the optimal solution to the packet mix optimization for a fixed set of transmitting devices by introducing the local IDNC graph for packet generation.

### A. Problem Formulation

Let  $\mathcal{A} \in \mathcal{P}(\mathcal{M})$  be the set of transmitting devices and let  $\mathcal{T}(\mathcal{A})$  be the set of non-transmitting devices in interference. In other words,  $\mathcal{T}(\mathcal{A})$  is the set of devices that can hear multiple transmissions from the set of transmitting devices  $\mathcal{A}$ . The mathematical definition of this set is:

$\mathcal{T}(\mathcal{A}) = \{i \notin \mathcal{A} \mid \exists (m, n) \in \mathcal{A}^2, m \neq n, i \in \mathcal{C}_m \cap \mathcal{C}_n\}$ . Define  $\mathcal{S}(\mathcal{A})$  as the set of devices that are not in the transmission range of any transmitting device. Formally, the set is written as:

$$\mathcal{S}(\mathcal{A}) = \{i \in \mathcal{M} \mid \nexists j \in \mathcal{A}, i \in \mathcal{C}_j\}. \quad (1)$$

Define the opportunity zone  $\mathcal{O}_i(\mathcal{A}) = \mathcal{C}_i \setminus (\mathcal{A} \cup \mathcal{T}(\mathcal{A}))$  as the set of devices that can be targeted by device  $i$  and can decode a packet from the transmission.

Let  $M_w$  be the set of devices having non-empty Wants set and let  $\kappa_i(\mathcal{A})$  be the packet combination to be transmitted by device  $i \in \mathcal{A}$ . Define  $\tau_i(\kappa_i(\mathcal{A}), \mathcal{A})$  as the set of targeted devices by device  $i$  when he is sending the packet combination  $\kappa_i$  and all devices in  $\mathcal{A}$  are transmitting.

**Remark 1.** *The variables defined above should be all a function of the set of transmitting devices  $\mathcal{A}$ . However, for notation convenience, the set will be dropped unless it is required (e.g., we should write  $\mathcal{T}$  and  $\tau_i(\kappa_i)$  instead of  $\mathcal{T}(\mathcal{A})$  and  $\tau_i(\kappa_i(\mathcal{A}), \mathcal{A})$ ).*

The minimum decoding delay problem can be formulated as a joint optimization problem over the set of transmitting devices and their packet combinations as illustrated in the following theorem:

**Theorem 1.** *The decoding delay reduction problem in partially connected D2D network can be formulated as follows:*

$$\begin{aligned} \max_{\mathcal{A} \in \mathcal{P}(\mathcal{M})} & -|\mathcal{A} \cap M_w| - |\mathcal{T} \cap M_w| - |\mathcal{S} \cap M_w| \\ & + \sum_{i \in \mathcal{A}} \left( \sum_{j \in \tau_i(\kappa_i^*)} 1 - p_{ij} \right) \end{aligned} \quad (2a)$$

subject to

$$\kappa_i^*(\mathcal{A}) = \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{H}_i)} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right), \forall i \in \mathcal{A}. \quad (2b)$$

*Proof:* The proof can be found in Appendix A. ■

Let the objective function (2a) of the optimization problem (2) be called the outer problem, and the constraint (2b) be called the inner problem. Finding the global optimal solution to the optimization problem (2) may require a search over all the sets of transmitting devices and their possible packet combinations. Such solution is clearly infeasible for any reasonable sized network. In the next subsection, the paper proposes finding the optimal packet mix for a fixed set of transmitting devices, i.e., the optimal  $\kappa_i^*$ ,  $i \in \mathcal{A}$  for a fixed set  $\mathcal{A}$ . In other words, the global optimal solution to the inner problem (2b).

### B. Local IDNC Graph for Packet Generation

Let the set of transmitting devices be fixed to  $\mathcal{A}$ . To solve the optimization problem (2b), this subsection rely on a graph theoretical model first introduced in the context of PMP networks in [12]. Therefore, this section first extends the IDNC graph to account for the restriction in the packet generation and the partial connection of the graph. In this context, let such graph be called the *local IDNC graph*. Finally, the paper reformulates the problem (2b) as a maximum weight clique problem in the local IDNC graph.

The IDNC graph is a tool introduced for the PMP model to determine both all possible XOR-based coded combinations, and the devices that can instantly decode each of them. In contract with PMP model that permit the generation of all the packet mixes, in D2D environment, each device can generate coding combinations only from the packets it possesses (i.e., packets in his Has set). Further, while the sender in the fully connected network can intend packets to all devices, in partially connected systems, each device can target only the devices in its transmission range. This subsection illustrates how the different devices can build similar local IDNC graphs. Naturally, the local graph depends on the set of transmitting devices. However following the simplification adopted in Remark 1, the set of transmitting users is dropped.

To construct the local IDNC graph  $\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$  of device  $i \in \mathcal{A}$ , a vertex  $v_{kl} \in \mathcal{V}_i$  is generated for each packet  $l \in (\mathcal{W}_k \cap \mathcal{H}_i)$ ,  $\forall k \in \mathcal{O}_i$ . An edge in  $\mathcal{E}_i$  connecting each vertices  $v_{kl}$  and  $v_{mn}$  is created if one of the two following conditions is true:

- $l = n \Rightarrow$  Packet  $l$  is needed by both devices  $k$  and  $m$ .
- $l \in \mathcal{H}_m$  and  $n \in \mathcal{H}_k \Rightarrow$  The packet combination  $l \oplus n$  is instantly decodable for both devices  $k$  and  $m$ .

The following lemma characterizes the solution of the decoding delay reduction problem (2b) for a fixed set of transmitting devices  $\mathcal{A}$ .

**Lemma 1.** *The optimal solution  $\kappa_i^*$  of the optimization problem (2b) for device  $i \in \mathcal{A}$  is the maximum weight clique in the local IDNC graph  $\mathcal{G}_i$  of device  $i$  in which the weight of each vertex  $v_{kl}$  is  $1 - p_{ik}$ .*

*Proof:* The proof can be found in Appendix B. ■

Given the solution to the decoding delay reduction problem (2b) for a fixed set of transmitting devices, the global optimal solution to the decoding delay reduction problem (2) can be obtained by solving problem (2b) for all possible set of transmitting devices. Since an exhaustive search over the set of devices  $\mathcal{M}$  is needed, the complexity of such algorithm is in the order of  $2^M \mathfrak{f}$ , where  $\mathfrak{f}$  is the complexity of solving (2b) (i.e., the complexity of solving the maximum weight clique problem). Therefore, such approach is clearly infeasible.

The rest of the paper shows an efficient method for reaching the global optimal solution of the optimization problem (2). In the next section, the optimization problem (2) is globally solved when imposing restrictions on the set of transmitting devices that cooperation is allowed only when no device is in interference. In other words, the problem is solved under the constraint  $\mathcal{T} = \emptyset$ . The section solves the problem

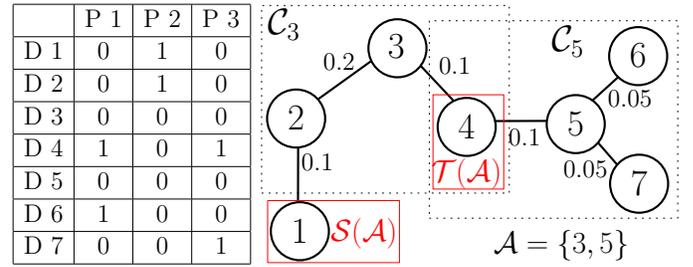


Fig. 1. Network composed of 7 devices and 3 packets. The feedback matrix represents the distribution of lost (1) and received packets (0) at each device. The erasure probabilities between devices is presented on the edges.

by introducing the *cooperation graph* and reformulating the problem as a maximum weight clique problem in that graph. Finally, the paper shows that employing a clustering approach and using both the cooperation graph and the local IDNC graph, the global optimal solution of (2) can be achieved.

### IV. INTERFERENCE-LESS COOPERATIVE SOLUTION

Due to the interdependence of the set of transmitting devices and transmitted packets, the variables are not separable as shown in Fig. 1 which presents a system and its associated feedback matrix.

Given the network configuration and the distribution of the lost/received packets, it can clearly be seen that only device 3 and device 5 can transmit a packet and ensure that it can be decoded by at least one device upon successful reception. When only device 3 is transmitting, the optimal solution is to target device 4 with packet 1 or 3. When only device 5 is sending, the optimal solution is to target both devices 6 and 7 with the packet combination  $1 \oplus 3$ . It can be easily shown that the optimal schedule is that both:

- device 3 targets device 2 with packet 2.
- device 5 targets devices 6 and 7 with the combination  $1 \oplus 3$ .

This solution shows the high interdependence between the variables. In this section, the decoding delay problem is relaxed by focusing on cooperation without interference. In other words, the cooperation between devices is allowed only when no device experiences interference (i.e.,  $\mathcal{T} = \emptyset$ ). This limitation makes the problem more mathematically tractable by allowing the decoupling of variables, which will serve as basis to solve the original problem. Hence, this section solves the following optimization problem:

$$\max_{\mathcal{A} \in \mathcal{P}(\mathcal{M})} -|\mathcal{A} \cap M_w| - |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} \left( \sum_{j \in \tau_i(\kappa_i^*)} 1 - p_{ij} \right) \quad (3)$$

subject to  
 $\mathcal{T} = \emptyset$

$$\kappa_i^*(\mathcal{A}) = \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{H}_i)} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right), \forall i \in \mathcal{A}.$$

To solve the optimization problem (3), this section first proposes reformulating the problem in a more tractable form. The rest of the section illustrates solve the problem by introducing the cooperation graph.

### A. Relaxed Problem Formulation

The newly introduced constraint  $\mathcal{T} = \emptyset$  limits the combinations of the transmitting devices. Let  $\mathcal{I}$  be the set of possible combinations of devices that satisfy the constraint. This set can be expressed as follows:

$$\mathcal{I} = \{\mathcal{A} \in \mathcal{P}(\mathcal{M}) \mid \mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \forall (i, j) \in \mathcal{A}^2\}. \quad (4)$$

The following theorem reformulates the optimization problem (3):

**Theorem 2.** *The decoding delay reduction problem in an interference-less IDNC-based D2D network (3) can be expressed as:*

$$A^* = \arg \max_{\mathcal{A} \in \mathcal{I}} -|\mathcal{A} \cap M_w| - |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} y_i(\kappa_i^*). \quad (5)$$

where

$$y_i(\kappa_i^*) = \max_{\kappa \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa)} 1 - p_{ij} \right). \quad (6)$$

*Proof:* The proof can be found in Appendix C. ■

As hinted in (5) and (6), the constraint  $\mathcal{T} = \emptyset$  allows the decoupling of the variables  $\mathcal{A}$  and  $\kappa_i$ ,  $i \in \mathcal{A}$ . Clearly, the optimization problem (6) is equivalent to the optimization problem (2b) that can be solved using the local IDNC graph. In the next subsection, the problem of selecting the set of transmitting devices (5) is solved using a graph theoretic formulation.

### B. Cooperation Graph

To solve the optimization problem (5), and equivalently the optimization problem (3), the paper characterizes the set of feasible device combinations  $\mathcal{I}$ . In general, two devices can transmit simultaneously if their coverage zones are mutually disjoint. Such combinations between the devices can be represented in a graph model called herein the *cooperation graph*.

The cooperation graph is constructed by creating a vertex  $v_i$  for each device in the network (i.e.,  $\forall i \in \mathcal{M}$ ). Two vertices  $v_i$  and  $v_j$  are connected by an edge if their coverage zone are disjoint. In other words, they are connected if  $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ . The following theorem characterizes :

**Theorem 3.** *The optimal solution to the relaxed decoding delay reduction problem in an interference-less IDNC-based D2D network (3) is the maximum weight clique problem in the cooperation graph, in which the weight of each vertex  $v_i$  is:*

$$v_i = |\mathcal{C}_i \cap M_w| - |\{i\} \cap M_w| + y_i(\kappa_i^*), \quad (7)$$

where  $\kappa_i^*$  is the solution to the maximum weight clique problem in the local IDNC graph illustrated in Lemma 1.

*Proof:* The proof can be found in Appendix D. ■

## V. COOPERATIVE DECODING DELAY REDUCTION

This section proposes to solve the decoding delay minimization problem (2) by introduction clustering of devices and using the tools developed in the previous sections. The fundamental concept for solving problem (2) is to generate groups of non-interfering devices (called herein clusters) and

to extend the cooperative graph formulation. Each cluster can be seen as a new “virtual” device in the network. Since the clusters are, by construction, non-interfering clusters then results of the previous section hold. Therefore, this section first shows the way of constructing such clusters then it extends the cooperative graph with the new virtual devices. Afterwards, the optimization problem (2) is shown to be equivalent to a maximum weight clique problem that can be efficiently solved using existing literature from graph theory [30], [31]. Finally, the paper demonstrate that generating only a well-defined subset of the available clusters is sufficient to reach the optimal solution of (2) with a lower complexity.

### A. Cluster Generation

Let  $\mathcal{A} \in \mathcal{P}(\mathcal{M})$  be a set of transmitting devices. This subsection illustrates the construction of a set of clusters  $\mathcal{Z}$ , called herein a clustering, that uniquely represents  $\mathcal{A}$  (i.e.,  $\mathcal{Z}$  is a partition of  $\mathcal{A}$ ). Each cluster  $Z \in \mathcal{Z}$  (i.e.,  $Z \subseteq \mathcal{A}$ ) is seen as a vertex in the cooperation graph and finally the uniqueness of the clustering  $\mathcal{Z}$  allows the reformulation of the problem in the next subsections.

Let  $\mathcal{Z}$  be a clustering (i.e., a partition) of  $\mathcal{A}$  such that:

- 1) All the coverage zones of clusters  $Z \in \mathcal{Z}$  are pairwise disjoint.
- 2) Within the same cluster  $Z \in \mathcal{Z}$ , each subset of devices is interfering with at least another device.

The mathematical definition of such a clustering  $\mathcal{Z}$  is:

$$\bigoplus_{Z \in \mathcal{Z}} Z = \mathcal{A} \quad (8a)$$

$$\mathcal{C}^T(Z) \cap \mathcal{C}^T(Z') = \emptyset, \forall Z \neq Z' \in \mathcal{Z} \quad (8b)$$

$$\mathcal{C}^T(z) \cap \mathcal{C}^T(Z \setminus z) \neq \emptyset, \forall z \subset Z, Z \in \mathcal{Z}, \quad (8c)$$

where  $\mathcal{C}^T(X)$  is the total coverage zone of all the devices in the set  $X$  defined as  $\mathcal{C}^T(X) = \bigcup_{x \in X} \mathcal{C}_x$ .

The following lemma states the uniqueness of the clustering  $\mathcal{Z}$  for any set of transmitting devices  $\mathcal{A}$ .

**Lemma 2.** *For any combination of transmitting devices  $\mathcal{A}$ , there exist a unique clustering  $\mathcal{Z}$  satisfying the constraints (8a), (8b) and (8c) simultaneously.*

*Proof:* The proof can be found in Appendix E. ■

### B. Full Cooperation Graph

In this subsection, the cooperation graph is extended to include the cluster that can be seen as “virtual” devices in the network. The fundamental concept in constructing such a graph is to preserve all the benefits of the cooperation graph while allowing all the possible combinations of transmitting devices. This goal is reached by generating particular clusters and then connecting each pair of clusters that are not interfering with each other. Let  $\mathbf{Z}$  be the set of clusters satisfying the constraint (8c) defined as follows:

$$\mathbf{Z} = \{Z \in \mathcal{P}(\mathcal{M}) \mid \mathcal{C}^T(z) \cap \mathcal{C}^T(Z \setminus z) \neq \emptyset, \forall z \subset Z\}. \quad (9)$$

The full cooperation graph is constructed by generating a vertex  $v$  for each cluster  $Z \in \mathbf{Z}$ . Two vertices  $v$  and  $v'$  representing the clusters  $Z$  and  $Z'$  are connected if they are

non-interfering clusters. In other words, they are connected if they satisfy constraint (8b):

$$\mathcal{C}^T(Z) \cap \mathcal{C}^T(Z') = \emptyset. \quad (10)$$

Let  $\mathfrak{Z}$  be the set of cliques in the full cooperation graph. The following proposition links the set of cliques to the set of transmitting devices.

**Proposition 1.** *There exists a one to one mapping between the set  $\mathcal{P}(\mathcal{M})$  and  $\mathfrak{Z}$ . In other words, for each  $\mathcal{A} \in \mathcal{P}(\mathcal{M})$  there exist a unique representative  $Z \in \mathfrak{Z}$  and inversely.*

*Proof:* The proof of this proposition is based on the result provided in Lemma 2. For a clique  $Z \in \mathfrak{Z}$  it is straightforward to conclude that there is a unique  $\mathcal{A} \in \mathcal{P}(\mathcal{M})$  that represent it. In fact, by the construction of the vertices  $Z$  they satisfy constraint (8c). Further since  $Z$  is a clique then all the vertices are connected, and hence they verify constraint (8b). For  $\mathcal{A} = \bigoplus_{Z \in \mathfrak{Z}} Z$  the mapping is unique. Conversely, let  $\mathcal{A}$  be a set of transmitting devices. From Lemma 2, there exist a unique  $Z$  that satisfy the constraints (8a), (8b), and (8c). Since all the cluster  $Z \in \mathfrak{Z}$  satisfy (8b) and (8c) then they are generated in the graph and they are connected. In other words,  $\mathfrak{Z}$  is a clique which concludes the proof. ■

### C. Decoding Delay Reduction

This subsection first reformulates the optimization problem (2) in a more tractable form based on the results provided in Proposition 1. Afterwards, the problem is shown to be equivalent to a maximum weight clique problem in the full cooperation graph. Finally, next section provides a low complexity solving method that relies on the generation of a well-defined set of clusters to construct the cooperation graph. Given the one to one mapping between the set of clique and the set of transmitting devices, the optimization problem (2) can be rewritten as follows:

$$\begin{aligned} & \max_{Z \in \mathfrak{Z}} -|Z \cap M_w| - |\mathcal{T} \cap M_w| - |\mathcal{S} \cap M_w| \\ & + \sum_{Z \in \mathfrak{Z}} \sum_{i \in Z} \left( \sum_{j \in \tau_i(\kappa_i^*)} 1 - p_{ij} \right) \end{aligned} \quad (11a)$$

$$\text{subject to } \kappa_i^*(Z) = \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{H}_i)} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right) \quad (11b)$$

The following theorem characterizes the global optimal solution to the decoding delay reduction in IDNC-based D2D network (2):

**Theorem 4.** *The optimal solution to the decoding delay reduction problem in partially connected IDNC-based D2D network (2) is the maximum weight clique problem in the cooperation graph, in which the weight of each vertex  $v$  representing the cluster  $Z$  is:*

$$\begin{aligned} v &= |\mathcal{C}^T(Z) \cap M_w| - |Z \cap M_w| - |\mathcal{T}(Z) \cap M_w| \\ & + \sum_{i \in Z} \left( \sum_{j \in \tau_i(\kappa_i^*)} 1 - p_{ij} \right), \end{aligned} \quad (12)$$

where  $\kappa_i^*$  is the maximum weight clique problem in the local

IDNC graph for packet generation:

$$\kappa_i^* = \arg \max_{\kappa \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa)} 1 - p_{ij} \right). \quad (13)$$

*Proof:* The proof can be found in Appendix F. ■

### D. Low Complexity Solving Method

This subsection presents a low complexity for solving the decoding delay minimization problem (2). As shown in the previous paragraph, the optimal solution can be represented as a maximum weight clique in the cooperation graph. This subsection answers to the question: Are all the clusters  $Z \in \mathfrak{Z}$  needed in the generation of the cooperation graph to achieve the optimal solution to the optimization problem? The paper answers the question by introducing a sequential low complexity method for constructing the cooperation graph by eliminating the vertices that are surely not part of the maximum weight clique.

Let  $Z \in \mathfrak{Z}$  be a cluster and let  $k \notin Z$  be a device such that:

$$\mathcal{C}_k \cap \mathcal{C}^T(Z) \neq \emptyset. \quad (14)$$

Let  $\mathfrak{Z} \in \mathfrak{Z}$  be a clustering (i.e., a clique in the cooperation graph) such that  $Z \in \mathfrak{Z}$ . First note that  $k \notin Z'$ ,  $\forall Z' \in \mathfrak{Z}$ . Otherwise, since  $k$  satisfy (14) then  $\mathfrak{Z}$  would violate (8b) and hence it is not a clique. This subsection first exhibits the condition under which the consideration of clusters that include device  $k$  would be beneficial. Afterwards, it proposes a method for generating only such clusters.

Let  $\mathfrak{Z}^k$  be a clustering, similar to  $\mathfrak{Z}$ , in which the cluster  $Z$  is replaced by the cluster  $Z^k = Z \cup \{k\}$ . In other words, the clustering  $\mathfrak{Z}^k$  can be written as  $\mathfrak{Z}^k = (\mathfrak{Z} \setminus Z) \cup \{Z \cup \{k\}\}$ . In order to determine if all the clusters are needed to achieve the optimal solution, the paper compute the different delays of  $\mathfrak{Z}$  and  $\mathfrak{Z}^k$ . The delay for a set of transmitting users  $\mathfrak{Z}$  can be expressed as the delay induced by  $Z$  and the delays induced by  $Z' \neq Z$  as illustrated in (11). Therefore, the difference in delay between  $\mathfrak{Z}$  and  $\mathfrak{Z}^k$  is the following:

$$\begin{aligned} \Delta &= |\mathcal{C}^T(Z) \cap M_w| - |Z \cap M_w| - |\mathcal{T}(Z) \cap M_w| \\ & + y(Z) - \left( |\mathcal{C}^T(Z^k) \cap M_w| - |Z^k \cap M_w| \right. \\ & \left. - |\mathcal{T}(Z^k) \cap M_w| + y(Z^k) \right) \\ & = -|(\mathcal{C}^T(Z^k) \setminus \mathcal{C}^T(Z)) \cap M_w| + |\{k\} \cap M_w| \\ & + |(\mathcal{T}(Z^k) \setminus \mathcal{T}(Z)) \cap M_w| + y(Z) - y(Z^k). \end{aligned} \quad (15)$$

where

$$y(Z) = \sum_{i \in Z} \left[ \max_{\kappa_i \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa_i)} (1 - p_{ij}) \right) \right] \quad (16)$$

Clearly, if the quantity  $\Delta$  defined in (15) is a positive number, then including device  $k$  in the clique do not provide any gain as it brings more interference than it serves devices. This concludes that such clustering  $\mathfrak{Z}^k$  is not the maximum weight clique since  $\mathfrak{Z}$  is a clique with a higher weight. Based on this observation, the rest of this subsection provide an efficient method for constructing the cooperation graph while

**Algorithm 1** Graph generation**Require:**  $C_i, \forall i \in M$ .Initialize  $k = 0$ .Construct  $\mathcal{G}^0(\mathcal{V}, \mathcal{E})$  using sectionInitialize  $t = \text{true}$ .**while**  $t = \text{true}$  **do**  Set  $t \leftarrow \text{false}$ .  Initialize  $\mathcal{G}^{k+1} = \emptyset$   **for all**  $v^k \in \mathcal{G}^k$  **do**    **for all**  $z^0 \in \mathcal{G}^0$  **do**      **if**  $w(\{v^k, z^0\}) \geq \max(w(v^k), w(z^0))$  **then**        Set  $t \leftarrow \text{true}$ .        Create vertex  $y^{k+1} = \{v^k, z^0\}$         Set  $\mathcal{G}^{k+1} \leftarrow \mathcal{G}^{k+1} \cup y$       **end if**    **end for**  **end for**  Set  $k \leftarrow k + 1$ .  **for all**  $y^k \in \mathcal{G}^k$  **do**    **for all**  $x \in \bigcup_{i=0}^k \mathcal{G}^i$  **do**      **if**  $\mathcal{C}^T(x) \cap \mathcal{C}^T(y^k) = \emptyset$  **then**        Create edge between  $y^k$  and  $x$       **end if**    **end for**  **end for****end while**Set  $\mathcal{G} = \bigcup_{i=0}^k \mathcal{G}^i$ 

avoiding, at maximum, generating the clusters that are surely not part of the maximum weight clique.

To construct the cooperation graph, first generate a vertex  $v_j^0$  for each device  $j \in \mathcal{M}$  in the network. Let such set of vertices be called the first layer of the graph. Two vertices  $v_i^0$  and  $v_j^0$  respecting the non-interference condition  $\mathcal{C}^T(v_j^0) \cap \mathcal{C}^T(v_i^0) = \emptyset$  are connected. Afterwards, the weight of each vertex  $v$ , representing the cluster  $Z$ , is computed as follows:

$$w(v) = |\mathcal{C}^T \cap M_w| - |Z \cap M_w| - |\mathcal{T} \cap M_w| + y(Z), \quad (17)$$

where the function  $y(\cdot)$  is defined in (16).

After this first step, the second layer of the graph is constructed by merging the clusters of the previous layer with the cluster of the first layer if the equality in (15) holds. In other words, for each pair of clusters  $v_j^0$  and  $v_i^0$  that are not connected (otherwise the clustering violates (8c)), two scenarios can be distinguished:

- $w(\{v_i^0, v_j^0\}) \geq \max(w(v_i^0), w(v_j^0))$ : generate cluster  $v_{ij}^1$  representing the cluster  $Z = \{i, j\}$ .
- $w(\{v_i^0, v_j^0\}) < \max(w(v_i^0), w(v_j^0))$ : the cooperation is not beneficial. The cluster is not created.

After the cluster generating phase, the newly created vertices are connected with the other ones that satisfy the non-interference constraint. The process is repeated for all layers of the graph. Hence for layer  $k + 1$ , combine all vertices  $v_i^0$  with vertices  $v_j^k$  if they are not connected and compute the weight of  $w(\{v_i^0, v_j^k\})$  to decide to generate or not the cluster. The steps of the algorithm are summarized in Algorithm 1.

**Theorem 5.** *Solving the maximum weight clique in the co-*

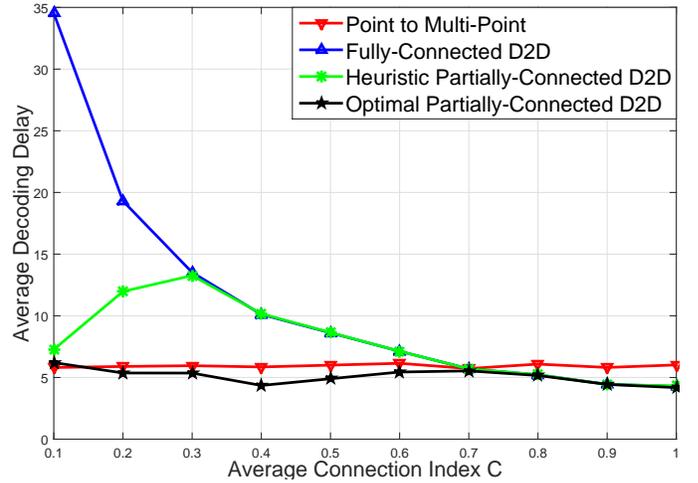


Fig. 2. Mean decoding delay versus the connectivity index  $C$  for a network composed of  $M = 60$  devices,  $N = 30$  packets, an erasure probability  $P = 0.1$ , and  $Q = 0.2$ .

operation graph generated by Algorithm 1 yields the optimal solution to the optimization problem (2). In other words, the maximum weight clique in the graph produced by Algorithm 1 is the same as the maximum weight clique in the cooperation graph constructed in Section V-B.

*Proof:* The proof can be found in Appendix G. ■

## VI. SIMULATIONS RESULTS

In this section, the simulation results comparing the decoding delay performance of the proposed schemes in the partially connected D2D system (problem (2) and the heuristic interference-less solution (3)) to both the conventional PMP system [12] (base station is responsible for both the *initial* phase and the *recovery* phase) and the solution proposed in [26] for fully connected network.

In these simulations, the decoding delay is computed over a large number of realizations, and the mean value is presented. Since the short range communications are more reliable than the base-station-to-device communications [25], [26], the D2D packet erasure probability  $P$  is set to be  $P = Q/2$  where  $Q$  is the base-station-to-device packet erasure probability. These erasure probabilities are assumed to be perfectly known for all devices.

Fig. 2 depicts the comparison of the average decoding delay achieved by the PMP configuration, the fully connected D2D policy, and our partially connected D2D optimal and heuristic policies against the connectivity index  $C$  for  $M = 60$ ,  $N = 30$ ,  $P = 0.1$  and  $Q = 0.2$ . From Fig. 2, we clearly see that for a low connectivity index ( $C \leq 0.3$ ) our partially connected approach largely outperforms the fully connected approach. This can be explained by the fact that in the fully connected D2D, only one device is transmitting at a time. For a highly connected graph, a single device can targeted all the others since the number of devices out of the transmission range is negligible. Whereas for a low connectivity index, this term is no longer negligible. Further, the heuristic solution clearly degrades as the connectivity index increase. This can be explained by the fact that as the connectivity index increases,

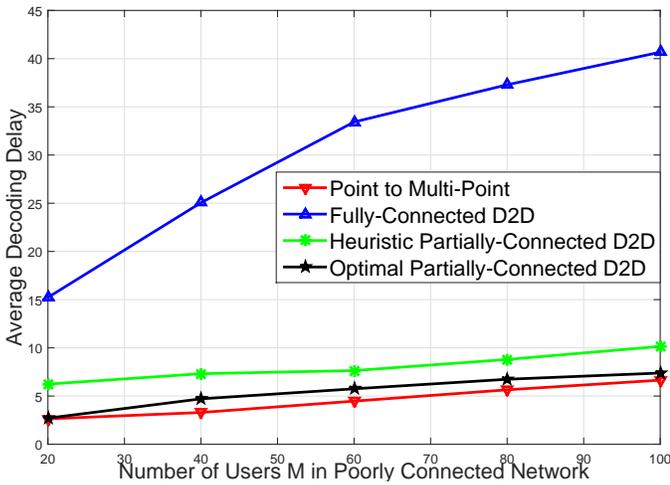


Fig. 3. Mean decoding delay versus number of devices  $M$  for a network composed of  $N = 30$  packets, a connectivity index  $C = 0.1$  an erasure probability  $P = 0.1$ , and  $Q = 0.2$

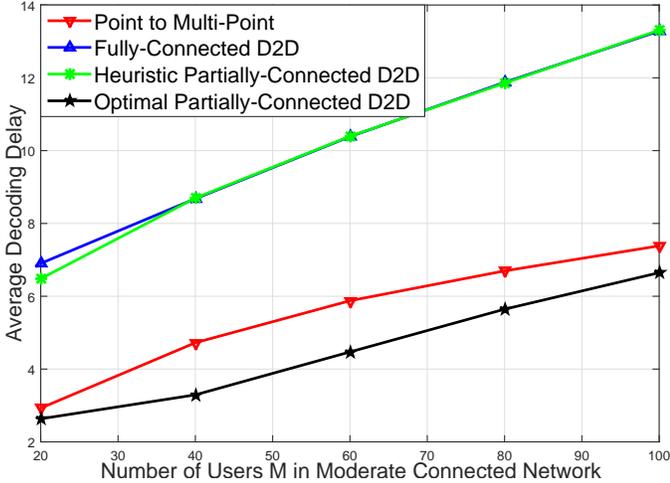


Fig. 4. Mean decoding delay versus number of devices  $M$  for a network composed of  $N = 30$  packets, a connectivity index  $C = 0.4$  an erasure probability  $P = 0.1$ , and  $Q = 0.2$ .

the number of devices that can transmit simultaneously decreases and thus the solution gets closer and closer to the fully connected D2D.

Fig. 3 and Fig. 4 illustrate the average decoding delay achieved by the same policies against the number of devices  $M$  for  $N = 30$ ,  $P = 0.1$ ,  $Q = 0.2$  for a poorly connected network ( $C = 0.1$  in Fig. 3) and a moderately connected network ( $C = 0.4$  in Fig. 4). Fig. 5 and Fig. 6 show a similar comparison against the number of packets  $N$  for  $M = 60$ ,  $P = 0.1$ ,  $Q = 0.2$  and  $C = 0.1$  in Fig. 5) and a moderately connected network ( $C = 0.4$  in Fig. 6).

For connectivity index  $C = 0.1$ , the heuristic partially connected and the optimal partially connected policies have the same performance as displayed in Fig. 3 and Fig. 5. This can be explained by the fact that at low connectivity, the set of devices that can transmit simultaneously while respecting the non-interference constraint is large and thus allowing interference do not provide a significant gain. Further, the partially connected algorithms provide an appreciable gain

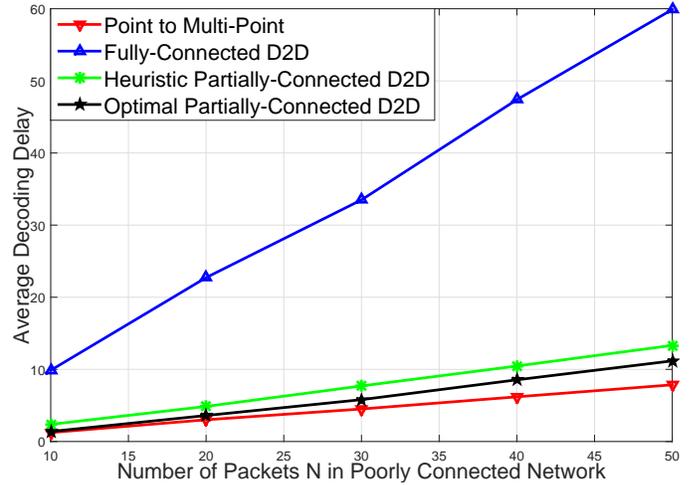


Fig. 5. Mean decoding delay versus number of packets  $N$  for a network composed of  $M = 60$  devices, a connectivity index  $C = 0.1$  an erasure probability  $P = 0.1$ , and  $Q = 0.2$ .

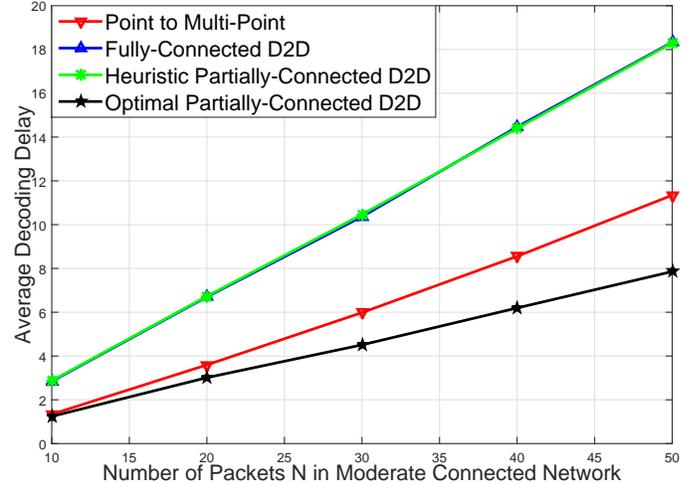


Fig. 6. Mean decoding delay versus number of packets  $N$  for a network composed of  $M = 60$  devices, a connectivity index  $C = 0.4$  an erasure probability  $P = 0.1$ , and  $Q = 0.2$ .

as compared with the fully connected solution as the number of devices and packets increases. For  $C = 0.4$ , we can clearly from Fig. 4 and Fig. 6 see that the optimal partially connected policy outperforms the PMP approach. Note that the PMP approach does not suffer from delay due to devices out of the transmitting range and delay encountered by the transmitting devices. As the connectivity index increases, these additional decoding delay decreases, which explain that the optimal D2D approach outperforms the PMP at these connectivity indexes.

Fig. 7 and Fig. 8 depict the same comparison for the decoding delay against the packet erasure probability  $P$  for  $M = 60$ ,  $N = 30$ ,  $Q = 2P$ ,  $C = 0.1$  for Fig. 7, and  $C = 0.4$  for Fig. 8. For a low connectivity index  $C$  our proposed solutions outperform the fully connected one for all values of erasure probabilities as shown in Fig. 7. As the connectivity index increases (Fig. 8), we clearly see that the performance of the optimal partially connected D2D is better than the PMP one. This can be explained by the fact that in our simulation, we fixed the base station-to-device erasure probability to be twice as large as the device-to-device erasure probability. For

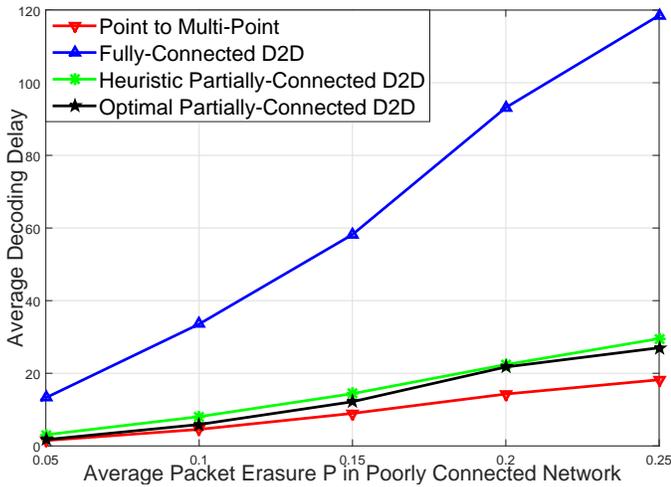


Fig. 7. Mean decoding delay versus erasure probability  $P$  for a network composed of  $M = 60$  devices,  $N = 30$  packets, a connectivity index  $C = 0.1$  an erasure probability  $Q = 2P$ .

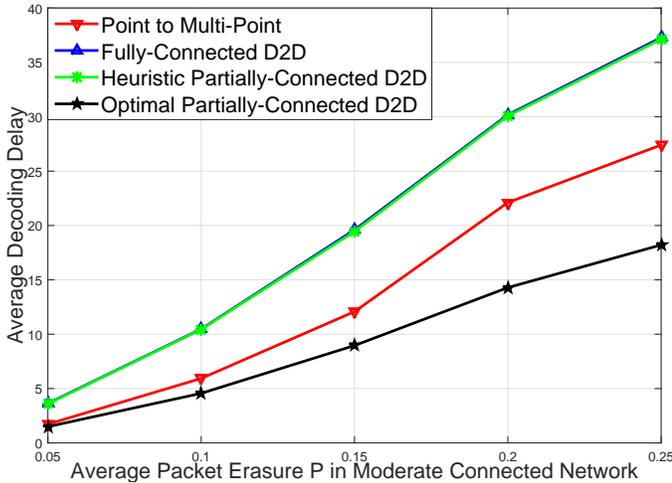


Fig. 8. Mean decoding delay versus erasure probability  $P$  for a network composed of  $M = 60$  devices,  $N = 30$  packets, a connectivity index  $C = 0.4$  an erasure probability  $Q = 2P$ .

a small erasure probability, the different between the D2D and the PMP erasure is not significant. As  $P$  increases,  $Q$  increases twice as much, which explain that the D2D approaches the PMP performance.

## VII. CONCLUSION

The paper investigates the decoding delay reduction problem in IDNC-based device-to-device communications network. In a D2D configuration, devices exchange encoded packets to help hasten the recovery of the lost packets of other devices in their transmission range. This paper considers a partially connected network in which more than one device can simultaneously transmit. The decoding delay minimization problem is formulated as a joint optimization over the set of transmitting devices and the packet combination to be transmitted. The paper first present a heuristic solution in which only non-interfering devices are allowed to transmit simultaneously. Afterwards, the paper proposes the global optimal solution to the decoding delay using a clustering approach. Both problems are shown to be equivalent to a

maximum weight clique problem in the constructed cooperation graph. Through extensive simulations, the decoding delay experienced by all devices in the Point to Multi-Point (PMP) configuration, the fully connected D2D (FC-D2D) configuration and the more practical partially connected D2D (PC-D2D) configuration are compared. Numerical results suggest that the PC-D2D outperforms the FC-D2D and provides appreciable gain especially for poorly connected networks.

## REFERENCES

- [1] A. Douik, S. Sorour, T. Y. Al-Naffouri, H.-C. Yang, and M.-S. Alouini, "Delay reduction in Multi-Hop Device-to-Device communication using network coding," in *Proc. of IEEE International Symposium on Network Coding (NetCod' 2015)*, Sydney, Australia, available Arxiv e-prints, vol. abs/1409.8460.
- [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, Jul 2000.
- [3] S. Sorour and S. Valaee, "On densifying coding opportunities in instantly decodable network coding graphs," in *Proc. of IEEE International Symposium on Information Theory (ISIT' 2012)*, Cambridge, MA, USA, July 2012, pp. 2456–2460.
- [4] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of IEEE International Symposium on Information Theory (ISIT' 2003)*, Yokohama, Japan, June 2003, pp. 442–448.
- [5] J.-S. Park, M. Gerla, D. Lun, Y. Yi, and M. Medard, "Codecast: a network-coding-based ad hoc multicast protocol," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 76–81, October 2006.
- [6] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments," in *Proc. of Annual Allerton Conference on Communication, Control and Computing (Allerton' 2005)*, Monticello, Illinois, USA, 2005.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM' 2006)*, Pisa, Italy, 2006, pp. 243–254.
- [8] 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 Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1094–1105, May 2011.
- [9] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM' 2009)*, Honolulu, Hawaii, USA, Nov 2009, pp. 1–6.
- [10] M. Esmailzadeh 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.
- [11] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Proc. of IEEE Information Theory Workshop (ITW' 2007)*, Bergen, Norway, Sept 2007, pp. 120–125.
- [12] 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.
- [13] S. Sorour, A. Douik, S. Valaee, T. Al-Naffouri, and M. Alouini, "Partially blind instantly decodable network codes for lossy feedback environment," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 4871–4883, Sept 2014.
- [14] M. Karim and P. Sadeghi, "Decoding delay reduction in broadcast erasure channels with memory for network coding," in *Proc. of IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC' 2012)*, Sydney, Australia, Sept 2012, pp. 60–65.
- [15] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proc. of Workshop on Network Coding, Theory, and Applications (NetCod' 2009)*, Lausanne, Switzerland, June 2009, pp. 80–85.

- [16] A. Douik, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Delay minimization for instantly decodable network coding in persistent channels with feedback intermittence," *Arxiv e-prints*, vol. abs/1004.1379, 2014.
- [17] L. Lu, M. Xiao, and L. K. Rasmussen, "Design and analysis of relay-aided broadcast using binary network codes," *Journal of Communications*, vol. 6, no. 8, pp. 610–617, 2011.
- [18] A. Le, A. Tehrani, A. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc of International Symposium on Network Coding (NetCod' 2013)*, Calgary, Canada, June 2013, pp. 1–6.
- [19] H. Tracey and S. Desmond, *Network Coding: An Introduction*. Cambridge, April, 2008.
- [20] 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, Dec. 2014.
- [21] A. Tehrani and A. Dimakis, "Finding three transmissions is hard," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM' 2012)*, Anaheim, California, USA, Dec 2012, pp. 2293–2298.
- [22] 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, June 2009, pp. 844–848.
- [23] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.
- [24] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, February 2014.
- [25] S. Tajbakhsh and P. Sadeghi, "Coded cooperative data exchange for multiple unicasts," in *Proc. of IEEE Information Theory Workshop (ITW' 2012)*, Lausanne, Switzerland, Sept 2012, pp. 587–591.
- [26] 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 (ISIT' 2013)*, Istanbul, Turkey, July 2013, pp. 3095–3099.
- [27] M. Karim, N. Aboutorab, A. Nasir, and P. Sadeghi, "Decoding delay reduction in network coded cooperative systems with intermittent status update," in *Proc. of IEEE Information Theory Workshop (ITW' 2014)*, Hobart, Tasmania, Australia, Nov 2014, pp. 391–395.
- [28] A. Douik, S. Sorour, H. Tembine, T. Y. Al-Naffouri, and M.-S. Alouini, "A game theoretic approach to minimize the completion time of network coded cooperative data exchange," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM' 2014)*, Austin, Texas, USA, Dec. 2014.
- [29] —, "A game-theoretic framework for decentralized cooperative data exchange using network coding," *submitted to IEEE Transactions on Mobile Computing*, available Arxiv e-prints, vol. abs/1004.1379, 2014.
- [30] P. R. J. Ostergard, "A fast algorithm for the maximum clique problem," *Discrete Appl. Math.*, vol. 120, pp. 197–207.
- [31] K. Yamaguchi and S. Masuda, "A new exact algorithm for the maximum-weight clique problem," in *Proc. Of the 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC' 2008)*, Yamaguchi, Japan.

## APPENDIX A PROOF OF THEOREM 1

In order to formulate the decoding delay minimization problem in partially connected IDNC-based D2D network, the expected decoding delay increase of each device is first expressed. Afterward, these expressions are used to formulate the problem as a joint optimization over the set of transmitting devices and the packet combinations to be transmitted.

From the definition of the decoding delay, device  $i$ , with non-empty Wants sets, experiences one unit increase of decoding delay if he cannot hear exactly one transmission or if he hears a packet that is either non-instantly decodable or non-innovative. The event of a device  $i$ , with non-empty Wants set, not be able to hear exactly one transmission occurs in the following scenarios:

- The device cannot hear any transmission: This event occurs if one of the following is true:

- The device is transmitting (i.e.,  $i \in \mathcal{A} \cap M_w$ ).
- The device is out of the transmission range of the transmitting devices (i.e.,  $i \in \mathcal{S} \cap M_w$ ).

- The device can hear multiple transmissions: This event occurs when the device is in the interference region of the transmitting devices (i.e.,  $i \in \mathcal{T} \cap M_w$ ).

The event of a device  $i$ , with non-empty Wants set, hearing a packet that is either non-instantly decodable or non-innovative happens when the device is in the opportunity zone of one of the transmitting devices (i.e.,  $i \in \mathcal{O}_j, j \in \mathcal{A}$ ) and one of the following event is true:

- He receives a non-innovative packet combination  $\kappa_j$ .
- He receives a non-decodable packet combination  $\kappa_j$ .

These last two events translate the fact that device  $i$  is not targeted by the transmission (i.e.,  $i \in (\mathcal{O}_j \setminus \tau_j(\kappa_j))$ ). Due to the erasure nature of the links, the probability that the event occurs is  $1 - p_j i$ .

In order to derive the expected decoding delay increase for an arbitrary device  $i$  in the network, we first partition the set of devices. From Lemma 3 (Appendix H), for any set of transmitting devices  $\mathcal{A}$ , the following sets forms a partition of the set of devices  $\mathcal{M}$ :

$$\mathcal{M} = \bigoplus_{i \in \mathcal{A}} (\mathcal{O}_i \cap M_w) \oplus (\mathcal{A} \cap M_w) \oplus (\mathcal{T} \cap M_w) \oplus (\mathcal{S} \cap M_w) \oplus \overline{M}_w. \quad (\text{A.1})$$

Given the partition of devices in (A.1) and the analysis above, the expected individual decoding delay  $d_j(\mathcal{A}, \kappa)$  of device  $j$  when devices in  $\mathcal{A}$  are transmitting respectively the packets combinations  $\kappa_i$  and  $\kappa = \{\kappa_i\}_{i \in \mathcal{A}}$  is given by:

$$\mathbb{P}(d_j(\mathcal{A}, \kappa) = 1) = \begin{cases} 0 & \text{if } j \in \overline{M}_w \\ 1 & \text{if } j \in \mathcal{A} \cap M_w \\ 1 & \text{if } j \in \mathcal{T} \cap M_w \\ 1 & \text{if } j \in \mathcal{S} \cap M_w \\ 0 & \text{if } j \in (\mathcal{O}_i \cap M_w) \cap \tau_i(\kappa_i) \\ 1 - p_{ij} & \text{if } j \in (\mathcal{O}_i \cap M_w) \setminus \tau_i(\kappa_i). \end{cases}$$

Let  $D(\mathcal{A}, \kappa)$  be the total decoding delay increase. From (A.1), all the sets form a partition of  $\mathcal{M}$  and thus are disjoint. Therefore, the expected overall decoding delay can be written as:

$$\begin{aligned} \mathbb{E}[D(\mathcal{A}, \kappa)] &= \sum_{i \in \mathcal{M}} d_i(\mathcal{A}, \kappa) \\ &= |\mathcal{A} \cap M_w| + |\mathcal{T} \cap M_w| + |\mathcal{S} \cap M_w| \\ &\quad + \sum_{i \in \mathcal{A}} \left( \sum_{j \in (\mathcal{O}_i \cap M_w) \setminus \tau_i(\kappa_i)} 1 - p_{ij} \right). \quad (\text{A.2}) \end{aligned}$$

Using the expected decoding delay increase expression (A.2), the decoding delay minimization problem can be formulated as the following joint optimization over the set of transmitting devices and the packet combination to be trans-

mitted:

$$\begin{aligned}
& \min_{\substack{\mathcal{A} \in \mathcal{P}(\mathcal{M}) \\ \kappa_i(\mathcal{A}) \in \mathcal{P}(\mathcal{H}_i)}} \mathbb{E} [D(\mathcal{A}, \kappa)] \\
&= \min_{\substack{\mathcal{A} \in \mathcal{P}(\mathcal{M}) \\ \kappa_i \in \mathcal{P}(\mathcal{H}_i)}} |\mathcal{A} \cap M_w| + |\mathcal{T} \cap M_w| \\
&\quad + |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} \left( \sum_{j \in (\mathcal{O}_i \cap M_w) \setminus \tau_i(\kappa_i)} 1 - p_{ij} \right) \\
&= \max_{\substack{\mathcal{A} \in \mathcal{P}(\mathcal{M}) \\ \kappa_i \in \mathcal{P}(\mathcal{H}_i)}} -|\mathcal{A} \cap M_w| - |\mathcal{T} \cap M_w| \\
&\quad - |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} \left( \sum_{j \in \mathcal{O}_i \cap \tau_i(\kappa_i)} 1 - p_{ij} \right) \\
&= \max_{\substack{\mathcal{A} \in \mathcal{P}(\mathcal{M}) \\ \kappa_i \in \mathcal{P}(\mathcal{H}_i)}} -|\mathcal{A} \cap M_w| - |\mathcal{T} \cap M_w| \\
&\quad - |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right). \quad (\text{A.3})
\end{aligned}$$

Apparently, from the problem formulation (A.3), the packet combination that the transmitting devices can generate only affects the last term of the expression. Therefore, the decoding delay minimization problem can be reformulated as follows:

$$\begin{aligned}
& \max_{\mathcal{A} \in \mathcal{P}(\mathcal{M})} -|\mathcal{A} \cap M_w| - |\mathcal{T} \cap M_w| - |\mathcal{S} \cap M_w| \\
&+ \sum_{i \in \mathcal{A}} \left( \sum_{j \in \tau_i(\kappa_i^*)} 1 - p_{ij} \right) \quad (\text{A.4})
\end{aligned}$$

subject to

$$\kappa_i^*(\mathcal{A}) = \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{H}_i)} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right), \quad \forall i \in \mathcal{A}. \quad (\text{A.5})$$

#### APPENDIX B PROOF OF LEMMA 1

To proof this lemma, an approach similar to the one employed in [18] for a PMP network is used. Since a device  $i$  can target only the devices in his opportunity zone  $\mathcal{O}_i$  and can make packet combination only using the packets it already holds, then the network can be reduced. In other words, considering the reduced network consisting of a set  $\mathcal{M}' = \mathcal{O}_i$  of devices and a set  $\mathcal{N}' = \mathcal{H}_i$  of packets yields the same solution. In fact, since  $\tau_i(\kappa_i) \subseteq \mathcal{O}_i$ , then the targeted devices in the original network are the same as in the reduced one. Moreover, given  $\kappa_i \subseteq \mathcal{H}_i$  then the packet combination remains unchanged. Therefore, the problem can be expressed as:

$$\kappa_i^* = \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{N}')} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right). \quad (\text{B.1})$$

According to the analysis done in [18], the optimization problem (B.1) is equivalent to a maximum weight clique in the IDNC graph [12] of the reduced network. Since the IDNC graph of the reduced network is equivalent to the local IDNC graph introduced in Section III, the result can be easily extended to the local IDNC graph. Therefore, the set of all

feasible packet combinations in local IDNC is represented by all maximal cliques in  $\mathcal{G}_i$ . To generate a packet combination, binary XOR is applied to all the packets identified by the vertices of a selected maximal clique  $\kappa$  in  $\mathcal{G}_i$ . The targeted devices by this transmission  $\kappa$  are those determined by the vertices of the selected maximal clique. In terms of the local IDNC graph, the optimization problem (2b) is equivalent to:

$$\begin{aligned}
\kappa_i^* &= \arg \max_{\kappa_i \in \mathcal{P}(\mathcal{H}_i)} \left( \sum_{j \in \tau_i(\kappa_i)} 1 - p_{ij} \right) \\
&= \arg \max_{\kappa_i \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa)} 1 - p_{ij} \right), \quad \forall i \in \mathcal{A}. \quad (\text{B.2})
\end{aligned}$$

where  $\kappa$  is a clique in the local IDNC graph. Therefore, the solution of the optimization problem (2b) for device  $i \in \mathcal{A}$  is the maximum weight clique in the local IDNC graph  $\mathcal{G}_i$  in which the weight of each vertex  $v_{jl}$  is  $1 - p_{ij}$ .

#### APPENDIX C PROOF OF THEOREM 2

To show the result, we first prove that the local IDNC graph is independent of the set of transmitting devices. This result, as shown in this proof, allows the decoupling of the variable of the problem (2), i.e., the decoupling of the problems (2a) and (2b).

From Lemma 4 (Appendix H), under the non-interference condition  $\mathcal{T} = \emptyset$ , the opportunity zone of any transmitting device  $i$  is independent of the set of transmitting devices. In other words, for any set of transmitting device  $\mathcal{A} \in \mathcal{I}$ :

$$\mathcal{O}_i(\mathcal{A}) = \mathcal{O}_i = \mathcal{C}_i \setminus \{i\}. \quad (\text{C.1})$$

The only variable that depends on the set of transmitting devices  $\mathcal{A}$ , in the construction of the local IDNC graph of device  $i$ , is the opportunity zone  $\mathcal{O}_i$ . Therefore, given that (C.1) holds under the non-interference constraint, then the local IDNC graph does not depend on the transmitting devices (i.e.,  $\mathcal{G}_i(\mathcal{A}) = \mathcal{G}_i$ ). As a consequence, the optimal packet combination device  $i$  can generate (optimization problem (2b)) do not depend on the set of transmitting devices as showed in following equation:

$$\begin{aligned}
\kappa_i^*(\mathcal{A}) &= \arg \max_{\kappa \in \mathcal{G}_i(\mathcal{A})} \left( \sum_{j \in \tau_i(\kappa)} 1 - p_{ij} \right) \\
&= \arg \max_{\kappa \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa)} 1 - p_{ij} \right) = \kappa_i^*, \quad \forall i \in \mathcal{A}. \quad (\text{C.2})
\end{aligned}$$

Let  $y_i(\kappa_i^*)$  be the value of the objective function of (C.2). Note that  $y_i(\kappa_i^*)$  is a non-negative function. The problem of selecting the transmitting devices can be expressed as:

$$\mathcal{A}^* = \arg \max_{\mathcal{A} \in \mathcal{I}} -|\mathcal{A} \cap M_w| - |\mathcal{S} \cap M_w| + \sum_{i \in \mathcal{A}} y_i(\kappa_i^*).$$

#### APPENDIX D PROOF OF THEOREM 3

To prove this theorem, we first reformulate the optimization problem (5) in a more tractable form. Afterward, we show that there is a one to one mapping between the set of feasible

transmitting devices and the cliques in the cooperation graph. Finally, since the weight of the clique is equivalent to the objective function of (5) then the optimal solution of (5) is the maximum weight clique in the cooperation graph.

Let  $\mathcal{C}^T(\mathcal{A}) = \bigcup_{i \in \mathcal{A}} \mathcal{C}_i$  be the total coverage zone of the transmitting devices. The set of devices, with non-empty Wants set, out of the transmission range of the transmitting devices can be written, under the non-interference constraint, as follows:

$$\begin{aligned} \mathcal{S} \cap M_w &= (\mathcal{M} \setminus \mathcal{C}^T(\mathcal{A})) \cap M_w = (\mathcal{M} \cap \overline{\mathcal{C}^T(\mathcal{A})}) \cap M_w \\ &= \overline{\mathcal{C}^T(\mathcal{A})} \cap M_w = (\overline{\mathcal{C}^T(\mathcal{A})} \cap M_w) \cup (M_w \cap \overline{M_w}) \\ &= M_w \cap (\overline{\mathcal{C}^T(\mathcal{A})} \cup \overline{M_w}) = M_w \cap (\overline{\mathcal{C}^T(\mathcal{A}) \cap M_w}) \\ &= M_w \setminus (\mathcal{C}^T(\mathcal{A}) \cap M_w). \end{aligned} \quad (\text{D.1})$$

Since  $\mathcal{T} = \emptyset$ , then we have  $\mathcal{C}^T(\mathcal{A}) = \bigoplus_{i \in \mathcal{A}} \mathcal{C}_i$ . Therefore, the cardinality of the set  $\mathcal{S} \cap M_w$  can be written as follows:

$$\begin{aligned} |\mathcal{S} \cap M_w| &= |M_w| - \left| \bigoplus_{i \in \mathcal{A}} \mathcal{C}_i \cap M_w \right| \\ &= |M_w| - \left| \bigoplus_{i \in \mathcal{A}} (\mathcal{C}_i \cap M_w) \right| \\ &= |M_w| - \sum_{i \in \mathcal{A}} |\mathcal{C}_i \cap M_w|. \end{aligned} \quad (\text{D.2})$$

Note that, the first term in (D.2) is constant with respect to the set of transmitting devices and the transmitted packet combinations. Thus, this term is ignored in the optimization problem. Clearly, we have  $|\mathcal{A} \cap M_w| = \sum_{i \in \mathcal{A}} |\{i\} \cap M_w|$ . Hence, the optimization problem (5) can be reformulated as follows:

$$A^* = \arg \max_{\mathcal{A} \in \mathcal{I}} \sum_{i \in \mathcal{A}} (|\mathcal{C}_i \cap M_w| - |\{i\} \cap M_w| + y_i(\kappa_i^*)) \quad (\text{D.3})$$

The rest of the section shown that there is a one to one mapping between the set of feasible transmitting devices (i.e.,  $\mathcal{A} \in \mathcal{I}$ ) and the set of cliques in the cooperation graph. Finally, to conclude the proof, we note that the weight of the clique is equivalent to the objective function of (D.3). We now prove that the solution of this problem is equivalent to the maximum weight clique in the cooperation graph where the weight of each vertex is:

Let  $\mathcal{A} \in \mathcal{I}$  be any combination of transmitting devices satisfying the non-interference constrain. We show that it can be represented by a clique  $\kappa$  in the cooperation graph. Let  $v_i$  be the vertices associated with the devices in  $\mathcal{A}$ ,  $i \in \mathcal{A}$ . By definition of  $\mathcal{I}$  we have  $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ ,  $\forall i \neq j \in \mathcal{A}$ . Therefore, vertices  $v_i$  and  $v_j$  are connected which concludes that  $\kappa$  is a clique in the cooperation graph.

In a similar way, let  $\kappa$  be a clique in the cooperation graph associated with the set of transmitting devices  $\mathcal{A}$ . Since all the nodes are pairwise connected then  $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ ,  $\forall v_i \neq v_j \in \kappa$  and hence  $\mathcal{A} \in \mathcal{I}$ . Therefore, there a one to one mapping between the set of feasible schedules  $\mathcal{I}$  and the set of cliques in the cooperation graph.

Finally, let  $\kappa$  be a clique. The weight can be expressed as

---

## Algorithm 2 Cluster Construction

---

**Require:**  $\mathcal{A}, \mathcal{C}_i, \forall i \in \mathcal{A}$ .

Initialize  $\mathcal{Z} = \emptyset$ .

Initialize  $k = 0$ .

**while**  $\mathcal{A} \neq \emptyset$  **do**

Set  $k \leftarrow k + 1$ .

Initialize  $Z_k = \emptyset$ .

Set  $Z_k \leftarrow \{a\}$ ,  $a \in \mathcal{A}$ .

Set  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$ .

Initialize  $t = \text{true}$ .

**while**  $t = \text{true}$  **do**

Set  $t \leftarrow \text{false}$ .

**for all**  $b \in \mathcal{A}$  **do**

**if**  $\mathcal{C}_b \cap \mathcal{C}^T(Z_k) \neq \emptyset$  **then**

Set  $Z_k \leftarrow Z_k \cup \{b\}$ .

Set  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{b\}$ .

Set  $t \leftarrow \text{true}$ .

**end if**

**end for**

**end while**

**end while**

Set  $\mathcal{Z} \leftarrow \{Z_1, \dots, Z_k\}$

---

follows:

$$\begin{aligned} w(\kappa) &= \sum_{v_i \in \kappa} w(v_i) \\ &= \sum_{v_i \in \kappa} (|\mathcal{C}_i \cap M_w| - |\{i\} \cap M_w| + y_i(\kappa_i^*)). \end{aligned} \quad (\text{D.4})$$

The weight in (D.4) is the objective function illustrated in (D.3). As a conclusion, the maximum weight clique in the cooperation graph is the optimal solution to the optimization problem (5) which conclude our proof.

## APPENDIX E PROOF OF LEMMA 2

To proof the uniqueness of the decomposition, this section first introduces an algorithm to perform the decomposition. Afterward, the rest of the section shows that the decomposition is unique.

To generate a clustering  $\mathcal{Z}$  associated with the set of transmitting devices  $\mathcal{A}$  respecting the conditions (8a), (8b) and (8c) simultaneously, this section proposes a sequential constructing method. A cluster  $Z_1$  is first generated by including an arbitrary element of the set  $\mathcal{A}$ . Afterward, all the remaining devices that are interfering with the cluster  $Z_1$  are added to it. It is easy to see that such method of construction ensures that constraint (8c) holds. After this step all the remaining devices in  $\mathcal{A}$  are not interfering with  $Z_1$ . A second cluster  $Z_2$  is generated by including one of the transmitting devices in  $\mathcal{A}$  and all the interfering devices added to it. Note that all the devices in  $Z_2$  are not interfering with devices in  $Z_1$ . Therefore,  $Z_1$  and  $Z_2$  are not interfering and by extension the clustering  $\mathcal{Z}$  satisfy the constraint (8b). The process is repeated until  $\mathcal{A} = \emptyset$  which ensures that constraint (8a) is satisfied. Therefore, the algorithm proves the existence of a clustering  $\mathcal{Z}$  respecting the conditions (8a), (8b) and (8c) simultaneously

for all combination of transmitting devices  $\mathcal{A}$ . The steps of the algorithm are summarized in Algorithm 2.

Now we proof uniqueness of such decomposition. Let  $\mathcal{A}$  be a set of transmitting devices and let  $\mathcal{Z}$  and  $\mathcal{Z}'$  be two distinct clustering of the set  $\mathcal{A}$  satisfying the constraints (8a), (8b) and (8c) simultaneously. From the constraint (8a), we have:

$$\bigoplus_{Z \in \mathcal{Z}} Z = A = \bigoplus_{Z' \in \mathcal{Z}'} Z'. \quad (\text{E.1})$$

Note that this condition implies only that the clustering have the same number of devices and not necessarily the same clusters  $Z$ . Since  $\mathcal{Z} \neq \mathcal{Z}'$ , given that both clustering satisfy (8a) then  $\exists i \in Z \cap Z'$  with  $Z \in \mathcal{Z}$ ,  $Z' \in \mathcal{Z}'$ ,  $Z \neq Z'$ .

Let  $K$  be the set constructed by first including the device  $i$ . Each device  $k$  in the set  $\mathcal{A}$  is included in  $K$  if  $\mathcal{C}_k \cap \mathcal{C}^T(K) \neq \emptyset$ . When no more devices can be added the process stop. Constraint (8b) can be written as:

$$\mathcal{C}^T(Z) \cap \mathcal{C}^T(\mathcal{A} \setminus Z) = \emptyset, \forall Z \in \mathcal{A}. \quad (\text{E.2})$$

From (8b), we can clearly see that  $K \subseteq Z$  and  $K \subseteq Z'$ . Assume that  $K \subset Z$ , then from (8c), we have:

$$K \subset Z \Leftrightarrow \mathcal{C}^T(Z \setminus K) \cap \mathcal{C}^T(K) \neq \emptyset. \quad (\text{E.3})$$

From the construction of the set  $K$ , we have:

$$\mathcal{C}^T(\mathcal{A} \setminus K) \cap \mathcal{C}^T(K) = \emptyset. \quad (\text{E.4})$$

Since  $\mathcal{C}^T(Z \setminus K) \subset \mathcal{C}^T(\mathcal{A} \setminus K)$ , then  $K = Z$ . The same reasoning can be done to  $Z'$  and thus  $Z = Z'$  and  $\mathcal{Z} = \mathcal{Z}'$ . This proves uniqueness of the decomposition and concludes our proof.

#### APPENDIX F PROOF OF THEOREM 4

To prove this theorem, we first develop each term of the optimization problem (11) in order to reformulate the problem in a more tractable form. Exploiting the constraint in the creation of the clusters  $Z \in \mathcal{Z}$ , we show that the problem can be formulated as an optimization over the individual clusters  $Z$  that can be seen as ‘‘virtual’’ devices in the network. Finally, using the results of Theorem 3, we conclude that the optimal solution to the optimization problem (11) is the maximum weight clique in the full cooperation graph.

Using an analysis similar to the one done in (D.1), the set of devices out of the transmission range of the transmitting devices can be written as:

$$\mathcal{S} \cap M_w = M_w \setminus (\mathcal{C}^T(\mathcal{Z}) \cap M_w). \quad (\text{F.1})$$

Since  $\mathcal{C}^T(Z) \cap \mathcal{C}^T(Z') = \emptyset, \forall Z \neq Z' \in \mathcal{Z}$  (constraint (8b) on the generation of the clusters), then we can write  $\mathcal{C}^T(\mathcal{Z}) = \bigoplus_{Z \in \mathcal{Z}} \mathcal{C}^T(Z)$ . Therefore, the cardinality of the set can be written as:

$$|\mathcal{S} \cap M_w| = |M_w| - \sum_{Z \in \mathcal{Z}} |\mathcal{C}^T(Z) \cap M_w|. \quad (\text{F.2})$$

As for (D.2), the first term in (F.2) is constant. Therefore, it is removed from the optimization problem. Clearly, we have  $|\mathcal{Z} \cap M_w| = \sum_{Z \in \mathcal{Z}} |Z \cap M_w|$ . Similarly, the set of devices in interference can be reformulated using the clusters  $Z$  as follows:

$$\mathcal{T}(\mathcal{Z}) = \bigcup_{Z, Z' \in \mathcal{Z}} \bigcup_{\substack{z \in Z \\ z' \in Z'}} (\mathcal{C}_z \cap \mathcal{C}_{z'}). \quad (\text{F.3})$$

From the constraint (8b), inter-cluster interference region is empty. In other words, we have:

$$\bigcup_{Z \neq Z' \in \mathcal{Z}} \bigcup_{\substack{z \in Z \\ z' \in Z'}} (\mathcal{C}_z \cap \mathcal{C}_{z'}) = \emptyset. \quad (\text{F.4})$$

Substituting (F.4) in (F.3) yields the following expression of the interference region:

$$\begin{aligned} \mathcal{T}(\mathcal{Z}) &= \bigcup_{Z \in \mathcal{Z}} \bigcup_{z \neq z' \in Z} (\mathcal{C}_z \cap \mathcal{C}_{z'}) \\ &= \bigoplus_{Z \in \mathcal{Z}} \mathcal{T}(Z). \end{aligned} \quad (\text{F.5})$$

Therefore, the cardinality of the set can be expressed as follows:

$$|\mathcal{T} \cap M_w| = \sum_{Z \in \mathcal{Z}} |\mathcal{T}(Z) \cap M_w|. \quad (\text{F.6})$$

Finally, the optimization problem (11) can be formulated as follows:

$$\begin{aligned} \mathcal{Z}^* &= \arg \max_{\mathcal{Z} \in \mathcal{Z}} \sum_{Z \in \mathcal{Z}} |\mathcal{C}^T(Z) \cap M_w| - |Z \cap M_w| \\ &\quad - |\mathcal{T}(Z) \cap M_w| + y(Z) \end{aligned} \quad (\text{F.7})$$

where the function  $y(Z)$  is defined as follows:

$$y(Z) = \sum_{i \in Z} \left[ \max_{\kappa_i \in \mathcal{G}_i} \left( \sum_{j \in \tau_i(\kappa_i)} (1 - p_{ij}) \right) \right] \quad (\text{F.8})$$

Since the clusters are non-interfering clusters, they can be seen as new *devices* in the network. Using a methodology very close to the one used in Theorem 3, the optimal clustering  $\mathcal{Z}^*$  is given by the maximal weight clique in the full cooperation graph where the weight of each vertex  $v$  representing the cluster  $Z$  can be expressed as:

$$v = |\mathcal{C}^T(Z) \cap M_w| - |Z \cap M_w| - |\mathcal{T}(Z) \cap M_w| + y(Z).$$

#### APPENDIX G PROOF OF THEOREM 5

To prove the theorem, it is sufficient to show that all the vertices in the maximum weight clique in the cooperation graph generated in Section V-B are created in the cooperation graph produced by Algorithm 1. This is because the cooperation graph produced by Algorithm 1 is a sub-graph of the full cooperation graph generated in Section V-B. Let  $\mathcal{Z}^* = \{Z_1^*, \dots, Z_{|\mathcal{Z}^*|}^*\}$  be the maximum weight clique in the cooperation graph generated in Section V-B. Let  $Z_k^* = \{z_1^k, \dots, z_{|Z_k^*|}^k\} \in \mathcal{Z}^*$  be a cluster where the element has been arranged such that:

$$\mathcal{C}_{z_i^k} \cap \mathcal{C}^T(\{z_1^k, \dots, z_{i-1}^k\}) \neq \emptyset, \forall 1 \leq i \leq |Z_k^*|. \quad (\text{G.1})$$

We prove by induction that all the clusters  $v_i = \{z_1^k, \dots, z_i^k\}, \forall 1 \leq i \leq |Z_k^*|$  are generated by Algorithm 1. For  $i = 1$ , the cluster (device)  $z_1^k$  is included in the graph by construction. Assume that cluster  $v_{i-1}$  is included. From the analysis done in (15), device  $z_i^k$  should provide a gain to be considered in the cluster. However, this result holds only for clusters that satisfy (14). We generalize the result for the clusters in the maximum weight clique. Note that the generalization is possible only for the maximum weight clique of the graph.

Assume that  $w(v_i) \leq w(v_{i-1})$ , then we have  $w(Z_k^*) \leq w(Z_k^* \setminus \{z_i^k\})$  and the cluster  $Z_k^* \setminus \{z_i^k\}$  is combinable with

all the clusters  $Z_i^*$ ,  $\forall 1 \leq i \neq k \leq |\mathcal{Z}^*|$  and hence  $\mathcal{Z} = \{Z_1^*, \dots, Z_{k-1}^*, Z_k^* \setminus \{z_i^k\}, Z_{k+1}^* \dots, Z_{|\mathcal{Z}^*|}^*\}$  will yields a higher weight. Since  $\mathcal{Z}^*$  is the maximum weight clique then we conclude that  $w(v_i) \geq w(v_{i-1})$ . The same proof can be done if  $w(v_i) \leq w(z_i^k)$  in which case  $\mathcal{Z} = \{Z_1^*, \dots, Z_{k-1}^*, Z_k^* \setminus \{v_{i-1}\}, Z_{k+1}^* \dots, Z_{|\mathcal{Z}^*|}^*\}$  will yields a higher weight. In other words, we have:

$$w(v_i) \geq \max(w(v_{i-1}), w(z_i^k)). \quad (\text{G.2})$$

Therefore, cluster  $v_i$  is created. This concludes that all the clusters in the maximum weight clique are generated in the cooperation graph resulting from Algorithm 1. All the connections are the same in both graphs. Therefore, the maximum weight clique in the cooperation graph created in Section V-B is the same as the one in the graph generated by Algorithm 1.

## APPENDIX H AUXILIARY RESULTS

### A. Partition of the Set of devices

**Lemma 3.** *For any set of transmitting devices  $\mathcal{A}$ , the following sets forms a partition of the set of all devices  $\mathcal{M}$ :*

$$\begin{aligned} \mathcal{M} = & \bigoplus_{i \in \mathcal{A}} (\mathcal{O}_i \cap M_w) \oplus (\mathcal{A} \cap M_w) \\ & \oplus (\mathcal{T} \cap M_w) \oplus (\mathcal{S} \cap M_w) \oplus \overline{M}_w. \end{aligned} \quad (\text{H.1})$$

*Proof:* We first prove that for any set of transmitting devices  $\mathcal{A}$ , the following results holds:

$$\mathcal{M} = \bigcup_{i \in \mathcal{A}} \mathcal{O}_i \cup \mathcal{A} \cup \mathcal{T} \cup \mathcal{S}. \quad (\text{H.2})$$

Let  $\mathcal{C}^T = \bigcup_{i \in \mathcal{A}} \mathcal{C}_i$  be the total coverage zone of the transmitting devices. By definition of the set  $\mathcal{S}$ , we have:

$$\mathcal{M} = \mathcal{C}^T \cup \mathcal{S}. \quad (\text{H.3})$$

Hence the opportunity zone can be written as follows:

$$\mathcal{O}_i = \mathcal{C}_i \setminus (\mathcal{A} \cup \mathcal{T}), \forall i \in \mathcal{A}. \quad (\text{H.4})$$

Developing the union of the opportunity zones yields:

$$\begin{aligned} \bigcup_{i \in \mathcal{A}} \mathcal{O}_i &= \bigcup_{i \in \mathcal{A}} \mathcal{C}_i \setminus (\mathcal{A} \cup \mathcal{T}) \\ \bigcup_{i \in \mathcal{A}} \mathcal{O}_i &= \mathcal{C}^T \setminus (\mathcal{A} \cup \mathcal{T}) \\ \mathcal{C}^T &= \bigcup_{i \in \mathcal{A}} \mathcal{O}_i \cup \mathcal{A} \cup \mathcal{T}. \end{aligned} \quad (\text{H.5})$$

Therefore, we obtain the desired result:

$$\mathcal{M} = \bigcup_{i \in \mathcal{A}} \mathcal{O}_i \cup \mathcal{A} \cup \mathcal{T} \cup \mathcal{S}. \quad (\text{H.6})$$

We now prove that the sets are disjoint. By definition of the sets  $\mathcal{A}$ ,  $\mathcal{T}$  and  $\mathcal{S}$ , we have that these sets are pairwise disjoint. Also by definition, we have the set  $\mathcal{O}_i$  is disjoint from  $\mathcal{A}$  and  $\mathcal{T}$ ,  $\forall i \in \mathcal{A}$ . Finally, assume  $\exists i \in \mathcal{O}_j \cap \mathcal{O}_k$  for  $(j, k) \in \mathcal{A}^2$ . We have  $\mathcal{O}_i \subset \mathcal{C}_i$ . Therefore,  $i \in \mathcal{C}_j \cap \mathcal{C}_k$ . By definition of the interference set  $\mathcal{T}$ , we have  $i \in \mathcal{T}$ . By definition of the opportunity zone, we have  $\mathcal{O}_i \cap \mathcal{T} = \emptyset$ ,  $\forall i \in \mathcal{A}$ . Hence such  $i \in \mathcal{O}_j \cap \mathcal{O}_k$  does not exist and all the sets  $\mathcal{O}_l, \forall l \in \mathcal{A}$  are pairwise disjoint. Obviously, we have  $(X \cap Y) \cap \overline{Y} =$

$\emptyset, \forall X, Y$ . This conclude our proof and we have:

$$\begin{aligned} \mathcal{M} = & \bigoplus_{i \in \mathcal{A}} (\mathcal{O}_i \cap M_w) \oplus (\mathcal{A} \cap M_w) \\ & \oplus (\mathcal{T} \cap M_w) \oplus (\mathcal{S} \cap M_w) \oplus \overline{M}_w. \end{aligned} \quad (\text{H.7})$$

### B. No Interference Opportunity Zone

**Lemma 4.** *For any set of transmitting device  $\mathcal{A} \in \mathcal{I}$  satisfying the non-interference constraint, we have:*

$$\mathcal{O}_i(\mathcal{A}) = \mathcal{O}_i = \mathcal{C}_i \setminus \{i\}. \quad (\text{H.8})$$

*Proof:* Let  $\mathcal{A} \in \mathcal{I}$  satisfying the non interference constraint, we have by definition of the interference set  $\mathcal{T}(\mathcal{A}) = \emptyset$ . Hence, the expression of the opportunity zone  $\mathcal{O}_i(\mathcal{A})$  of any device  $i \in \mathcal{A}$  becomes:

$$\begin{aligned} \mathcal{O}_i(\mathcal{A}) &= \mathcal{C}_i \setminus (\mathcal{A} \cup \mathcal{T}(\mathcal{A})) \\ &= \mathcal{C}_i \setminus \mathcal{A}. \end{aligned} \quad (\text{H.9})$$

We now prove that  $(\mathcal{C}_i \setminus \{i\}) \cap \mathcal{A} = \emptyset$ . Assume  $\exists j \in (\mathcal{C}_i \setminus \{i\}) \cap \mathcal{A}$ . Then, by definition of the coverage zone, we have  $j \in \mathcal{C}_j$  and thus  $j \in \mathcal{C}_j \cap (\mathcal{C}_i \setminus \{i\}) \in \mathcal{T}(\mathcal{A})$ . However, by definition of the set  $\mathcal{A}$ , the interference region is empty  $\mathcal{T}(\mathcal{A}) = \emptyset$ . Therefore, such  $j \in (\mathcal{C}_i \setminus \{i\}) \cap \mathcal{A}$  does not exist, which conclude our proof and we have:

$$\mathcal{O}_i(\mathcal{A}) = \mathcal{O}_i = \mathcal{C}_i \setminus \{i\}. \quad (\text{H.10})$$