IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 9, 2019, accepted February 11, 2019, date of publication February 22, 2019, date of current version March 12, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2900982*

# A Coalitional Game-Theoretic Framework for Cooperative Data Exchange Using Instantly Decodable Network Coding

**MARIEM ZAYENE**[1,2], **OUSSAMA HABACHI**[2], **VAHID MEGHDADI**[2], **TAHAR EZZEDINE**[1], **AND JEAN PIERRE CANCES**[2]
[1]Communications Systems Laboratory (SysCom), National Engineering School of Tunis, Tunis 1002, Tunisia
[2]XLIM, University of Limoges, 87060 Limoges, France

Corresponding author: Oussama Habachi (oussama.habachi@unilim.fr)

**ABSTRACT** This paper investigates the cooperative data exchange (CDE) scheme using the instantly decodable network coding across energy-constrained devices over the wireless channels. In fact, enabling the CDE brings several challenges, such as how to extend the network lifetime and how to reduce the number of transmissions in order to satisfy the urgent delay requirements. The problem is modeled using the cooperative game theory in partition form. Unlike most existing studies, which are only delay-sensitive, we take into account both the completion time and the consumed energy. We propose a distributed merge-and-split algorithm to allow the wireless nodes to self-organize into the independent disjoint coalitions in a distributed manner. Indeed, the proposed algorithm guarantees reduced energy consumption and minimizes the delay in the resulting clustered network structure. Note that we have considered not only the transmission energy but also the computational energy consumption. Moreover, we focus on the mobility issue and we analyze how, in the proposed framework, the nodes can adapt to the dynamics of the network. Such an important result offers insights into how to design the scalable energy and delay aware CDE framework. The simulation results validate the proposed framework and show that, interestingly, the nodes reduce both the energy consumption and the completion time.

**INDEX TERMS** Coalitional game theory, cooperative data exchange, instantly decodable network coding, unmanned aerial network.

## I. INTRODUCTION

During the last decade, we have witnessed a proliferation in types and numbers of wireless devices and the ubiquity of wireless networks. Machine type communications (MTC) have emerged as a new communication paradigm that concerns devices, machines, and equipment that communicate with each other barely without human interventions. There have been a steady increase in the number of MTC devices, which are predicted to be in billions by 2020 [1]. Hence, there is an increasing need for innovative solutions at both system and device levels in order to address both environmental and operational costs. Consider a number of geographically distributed wireless nodes that are interested in receiving a set of packets broadcasted by a base station (BS). In such scenario, communication conditions are unreliable due to the high mobility of nodes. Hence, users are subject to packet losses. Moreover, the packet loss probability depends on the scenario and the environmental conditions, and are different from node to node. Indeed, in some scenarios such as drone fleets and VANET, mobile nodes may be out of the range of the BS when trying to recover the missed packets. Instead of relying on the BS to recover the dropped packets, an interesting strategy is the cooperation among the nodes by exchanging network coded packets until all of them have received all required packets. This configuration is called cooperative data exchange (CDE) [2]. Indeed, CDE is considered as a future research direction for several applications, owing to its several benefits such as the reduction of the load of the BS and the scalable and reliable delivery of the information. In this paper, we design a game theoretical communication framework for MTC networks that allows devices to use CDE to share information.

Specifically, we are interested in real-time energy efficient applications having two main characteristics: they have strict and urgent deadlines delay, and they are powered using

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

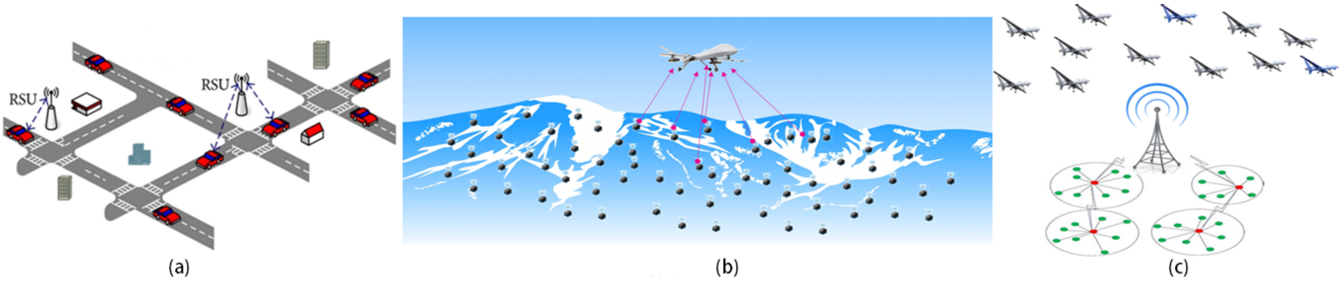

**FIGURE 1.** **Three use cases of our framework: a) Roadside base station broadcasting data to moving vehicles, b) A deployed WSN in a hostile environment, c) A drone fleet collecting information from the sink of a WSN.**

limited capacity batteries. The proposed framework can be applied in several scenarios as we can see in figure 1. We can consider either mobile nodes or mobile BSs. In fact, when a roadside base station is broadcasting data to vehicles, they can miss some packets due to their high-speed mobility. Thus, the proposed framework will be useful for vehicles to recover the missing packets. This framework is also interesting for a drone fleet that collects information from the sink of wireless sensor network. Indeed, during the last decade, the use of a multi-unmanned aerial vehicles (UAVs) system to cooperatively monitor a given area has been regularly increased, and has overcome the interest of using a single drone [3]–[5]. In such systems, small UAVs can autonomously cooperate, make decisions and take actions in order to meet the objectives of a particular mission [3], [4]. Another important application is a wireless sensor network (WSN) deployed in a hostile environment where sensors have to accumulate and store the same sensed data until the visit of the mobile sink.

Network coding (NC) has been initiated in the seminal work [6] and since then, it has shown the potential abilities to enhance network efficiency due to the reduced number of transmitted packets when packet loss occurs. There are two main network coding schemes: the random network coding (RNC) [7] and the opportunistic network coding (ONC). The former allows the combination of all source packets using random coefficients, while in the latter, the coding node detects coding opportunities and exploits them to combine the appropriate packets. In our work, we are interested in a promising subclass of ONC, which is suitable for the aforementioned applications, called instantly decodable network coding (IDNC) [8], [9]. Indeed, IDNC gained much attention in recent years thanks to its attractive properties. In fact, IDNC is quite simple; every encoded packet is produced through a simple XOR operation and it is instantly decodable at multiple receivers upon successful reception. Moreover, no buffers are used at the nodes to save coded packets for future decoding possibilities.

In addition to the network efficiency, network robustness can be significantly improved by allowing the nodes in a network to mix different packets through algebraic combinations. Interestingly, in the CDE configuration, only network-coded packets are exchanged. This setting is more robust comparing to a noncooperative and centralized scheme with a single sender holding all packets and passive clients having several demands. However, in such configuration, the exchanged coded packets might be either intercepted to obtain information about main messages or modified by attackers. Recently, there exist several studies that have addressed these problems in the CDE-based network coding scheme, such in [10].

The remainder of this paper is organized as follows. Section II discusses the related works and the contributions. The system model and the new IDNC recovery protocol are described in Section III. Then, we present in Section IV the coalitional game model and the utility function. The description of the merge-and-split algorithm is provided in section V. Section VI compares and analyzes the performance of the proposed scheme, and Section VII concludes the paper.

## II. RELATED WORKS AND CONTRIBUTIONS

During the last decade, there has been a general focus on the investigation of completion time, decoding delay, delivery time and rate optimization for various IDNC applications [11]–[17]. Most of the proposed solutions are deployed in centralized setting where there exists a central unit such a (BS) that is in charge of the recovery of the missing packets. In [11], the problem of minimizing the completion delay is considered for IDNC in wireless multicast and broadcast scenarios. Since finding the optimal policy using stochastic shortest path is intractable, they designed a maximum weight clique selection algorithm with a reduced complexity. In [12], authors considered the problem of minimizing the completion time through decoding delay control in both scenarios: perfect and imperfect feedback over persistent erasure channels. They formulated the problem as a maximum weight clique problem in the IDNC graph and proposed two heuristic algorithms to solve the problem. On the other hand, in a wireless point-to-multipoint network, Karim *et al.* [13] considered the problem of reducing video distortion of a set of devices that are interested in receiving in realtime a video sequence broadcasted from a BS. Since finding the optimal solution using Markov decision process is intractable, they proposed an online maximal clique selection algorithm over a rate aware IDNC graph in order to select the suitable packet combination and transmission rate. For a content-aware IDNC in device-to-device (D2D) networks, where not all devices are interested in the same quality of content, Keshtkarjahromi *et al.* [14] proposed a novel content and loss aware IDNC scheme that improves jointly the completion time and content quality. Furthermore, Douik *et al.* [15] introduced a non-cooperative game theoretic framework in (D2D) networks in order to solve the IDNC delay minimization problem. They proposed three games to reduce the completion time, the maximum decoding delay and the sum decoding delay. Since they studied an MTC network similar to the one considered in this paper, we compare in Section VI the proposed framework with this non-cooperative solution and we show that cooperation is well-suited for the IDNC completion time minimization problem.

As one can clearly see, previous studies exploit the benefits of the network coding in order to satisfy numerous requirements of real-time applications. However, there are several applications where energy consumption metric must be of high concern in addition to the realtime requirements. For example, in a CDE scheme that involves energy-constrained devices, an optimal IDNC packet combination can target a device that optimizes the transmission rate or the delay. However, does the selected sender have enough stored energy to target any decoding receiver in the field? Can he really reach all receivers? Douik and Sorour [16] considered the joint optimization problem of selecting the transmitting user and file combination in a partially connected network in which devices are not all in the transmission ranges of each other. However, they considered only the problem of reducing the completion time needed to disseminate all files. Furthermore, the recovery process is performed by a central coordinator in the network not in a distributed manner.

Hence, we found a great interest to develop a framework that involves constrained-energy MTC devices in a CDE network using IDNC. Motivated by improving the network efficiency and modeling aspect for a fully autonomous wireless nodes, we design the CDE among wireless nodes using cooperative game theoretical framework in partition form. Moreover, we propose a distributed merge-and-split algorithm that creates appropriate coalition groups accounting to the completion time and energy efficiency. The major contributions are summarized as follows:

- We introduce a novel framework from coalitional game theory to model the cooperation in the IDNC game among nodes for energy efficient CDE.
- We focus jointly on the completion time and the consumed energy to increase the network lifetime.
- We propose a merge-and-split algorithm, which iteratively operates the coalition formation process in a distributed fashion, and show that it converges to a stable coalition network structure.
- The proposed framework is of low complexity compared to the non-coalitional model, especially for high number of nodes, which increases the scalability of the proposed model.
- We evaluate the proposed framework using two practical scenarios: A Wireless sensor network and a network of flying fleet of drones.
- We reduce the completion delay by considering additional constraint, i.e. the energy consumption. Indeed, we illustrate that the proposed framework reduces the energy consumption and the completion delay at the same time.

## III. SYSTEM MODEL AND DEFINITIONS

### A. SYSTEM MODEL AND RECOVERY PROTOCOL

We consider a BS trying to deliver a frame $\mathcal{N}$ of $N$ source packets $\{1, \cdots, N\}$ to a group $\mathcal{M}$ of $M$ wireless nodes, denoted $\{1, \cdots, M\}$, each of which requires the reception of all source packets. Note that the wireless nodes can be arranged in a unique cluster or in multiple clusters. The first source sender can be a simple node as it can be a wireless base station. Node $k \in \mathcal{M}$ may lose a packet from node $l \in \mathcal{M}$ with a probability $q_{k,l}$ that depends mainly on the distance between them. In this model, we assume the BPSK modulation is used in the physical layer transmission. The bit error probability is defined using the Q-function $P_b = Q(\sqrt{\delta})$, where $\delta$ represents the signal to noise ratio (SNR). $\delta \cong \frac{SNR_0}{d^{\beta}}$, where $d$ is the inter-node transmission distance, and $\beta$ is the path loss exponent. Thus, the packet erasure probability is given by $p = 1 - (1 - P_b)^L$, where $L$ is the number of data bits per packet.

At the beginning, the BS transmits sequentially the $N$ uncoded packets of the frame. For each successfully received packet, each user sends an acknowledgement to the BS. The retransmission of the packet is required only if it is not received by no user. Therefore, when at least each packet is acknowledged once, this initial phase ends. We assume that all transmission feedbacks are perfect.

For every node $k \in \mathcal{M}$, packets from $\mathcal{N}$ belong to one of the two following sets :

- The HAS set ($H_k$): packets successfully received by node $k$.
- The WANTS set ($W_k$): missed packets for node $k$.

The feedback matrix $L = [l_{k,j}]$, $k \in \mathcal{M}, j \in \mathcal{N}$, is expressed as follows:

$$l_{kj}(t) = \begin{cases} 0 & \text{if } j \in H_k \\ 1 & \text{if } j \in W_k \end{cases} \tag{1}$$

Before starting the recovery phase, nodes may arrange themselves by forming a novel partition $\Pi$ of collaborating nodes. A partition $\Pi$ is defined as a set $\{S_1, \cdots, S_m\}$ of $m$ mutually disjoint clusters such that $\bigcup_{i=1}^{m} S_i = \mathcal{M}$, with $m$ cluster heads $\mathcal{CH} = \{CH_1, \cdots, CH_m\}$. Let us denote by $\Pi'$ a partition with a unique cluster 'grand coalition'. Hence, this phase is performed under the control of coalition heads $\mathcal{CH}$ according to our proposed delay and energy aware IDNC cooperative scheme which will be introduced in the next section.

Subsequently, once the coalition formation phase is finished, the recovery phase begins. It consists of two successive subphases:

1) Intra-cluster recovery phase: In this sub-phase, nodes in the same coalition may cooperate to recover their

missing packets. In fact, at every time slot $t$ one sender is selected to transmit a binary XOR encoded packet by exploiting the diversity of its HAS sets and the received feedbacks from the remaining cluster members. The process is repeated until all cluster members recover all missing packets. However, it may happen that in a given cluster not all the $N$ packets are available. In such case, the inter-cluster recovery process begins. During this phase, since the sender targets only its cluster members, we assume that the same transmission frequency can be reused in different clusters at the same time. Note that spatial frequency reuse have been extensively investigated in [18].

2) Inter-cluster recovery phase: Only cluster heads $\mathcal{CH}$ perform this phase. After finishing their intra-cluster recovery phase, they cooperate with each other to recover the remaining packets. We assume that $\mathcal{CH}$ broadcast immediately the decoded packet to their coalition members. This process is repeated until all cluster heads report that they obtained all the packets.

Note that the Inter-cluster recovery phase is required only in the case of the non-availability of the $N$ packets in at least one coalition.

We assume that single hop transmissions are used within the clusters. Furthermore, the packets sets (feedbacks) of each user are known by all the other cluster members since they can overhear each-other's feedbacks. Indeed, maintaining a feedback matrix of the cluster is of lower complexity and of lower overhead than the non-coalitional model.

*Example 1:* Let us consider an example of a schedule of a clustered IDNC-network, illustrated in figure 2, where 6 devices are arranged into two clusters each of which is trying to recover a set of 3 packets $\{p_1, p_2, p_3\}$. In the intra-cluster recovery phase, devices in $S_1$ receive all their wanted packets after exchanging two network-coded packets $p_2 \oplus p_3$ and $p_1$. However, since no device in $S_2$ has the packet $p_3$, their intra-cluster recovery phase is blocked after their first recovery transmission. In that case, the inter-cluster recovery phase is required so that all devices in the second cluster receive $p_3$.

Note that the cluster heads designation is beyond the scope of this paper. In fact, many works have studied methods of cluster head selection (see [19]–[21]). In our proposed solution, the node that has more residual energy is supposed to be elected as a cluster head.

At each recovery stage, the suitable decoding packet in the cluster with its corresponding sender is selected by the CH taking into account the completion time and the consumed energy. In this setup, although the same frequency is reused throughout the clusters, the interference between nodes is reduced since the intra-recovery transmissions are made over short ranges. Nevertheless, in some cases where we have some scattered coalitions, collisions may happen during the recovery process. Moreover, executing the intra-cluster recovery phase simultanously in every cluster reduces significantly the time of recovery process. However, in [15], the delay-aware decision making of the suitable combined packets and the transmitting device is made by all nodes in the network and the network-coded packets are transmitted for all nodes whatever the network size is.

A packet received by node $k$ can be one of the following:

- **Non-innovative** if it does not bring new packet to the receiver.
- **Instantly Decodable** if it contains exactly one source packet from $W_k$.
- **Non-Instantly Decodable** if it contains two or more source packets from $W_k$.

*Example 2:* Let us consider again figure 2. $p2 \oplus p3$ is instantly decodable for $k2$ and $k3$ since $p2 \in H_3$ and $p3 \in H_2$. However, if $k2$ broadcasts in the first time slot $p1 \oplus p3$, it will be non-instantly decodable for $k3$ since $\{p1, p3\} \in W_3$, but it is instantly decodable for $k1$ since $p1 \in H_1$.

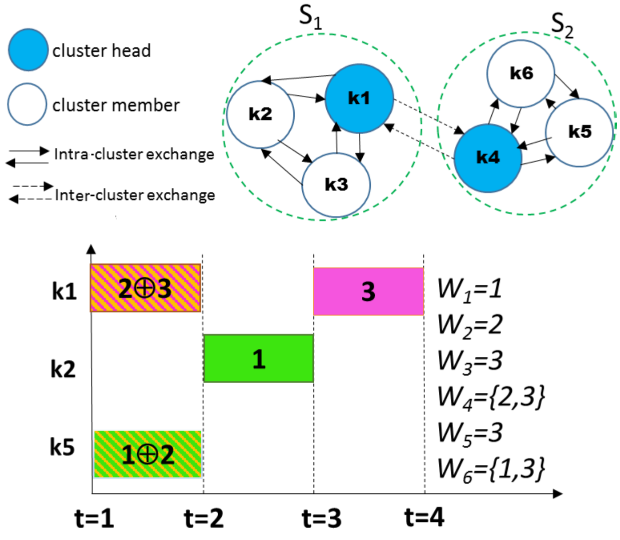

**FIGURE 2.** An example of a schedule in a network composed of 6 devices arranging into two clusters $S_1$ and $S_2$ where the clustered-CDE using IDNC is enabled; in the first time-slot $k1$ and $k5$ transmit at the same time, each of which in its cluster. The intra-cluster recovery phase of $S_2$ is blocked at $t = 2$ since packet 3 is unavailable. Members of $S_1$ finish recovering their missing packets after the second transmission. At $t = 3$, the inter-cluster recovery phase begins involving only cluster-heads $k1$ and $k4$.

### B. DEFINITIONS

This subsection presents all the definitions of cluster-IDNC delays. In addition to energy consumption minimization, the paper aims to minimize the number of required recovery stages in every cluster $S_i$ called the cluster-completion time $C_{S_i}$ defined as follows:

*Definition 1: For node k, the individual completion time $C_{k \in S_i}$ is the required number of recovery transmissions in such a way all its missing packets are received. Thus, the cluster-completion time $C_{S_i}$ is the total number of needed transmissions by cluster $S_i$ so that all its members recover their packets i.e. $C_{S_i} = \max_{k \in S_i} C_k$.*

Inspired by the study in [12], we consider the approach of decoding delay control in order to reduce the completion time. To re-express the cluster-completion time, let us first define the decoding delay. Let $t$ denote the time slot index or the recovery stage index when one node in every cluster performs a recovery transmission. For example, $t = 2$ refers to the second transmission.

*Definition 2: In each cluster $S_i$, a node $k \in S_i$, with non-empty $W_k$, encouters one unit increase of decoding delay, denoted by $d^t_{k\in S_i}$, if it receives a non-innovative or non-instantly decodable packet or if it does not receive any decoding packet. This can happen when the recovery process is stopped (due to the non-availability of the $N$ packets in the HAS sets of the cluster members) in that cluster waiting for the execution of the inter-cluster recovery phase.*

*Definition 3: In each cluster $S_i \in S$, the accumulative decoding delay $D^t_{k\in S_i}$ is the summation of the decoding delays units experienced by receiver $k$ until the time slot $t$. Thus, the overall decoding delay $D_{k\in S_i}$, experienced by $k$, is the summation of the decoding delays units throughout both recovery phases.*

*Corollary 1: The overall decoding delay experienced by node $k \in S_i$ is expressed as follows:*

$$D_{k\in S_i} = \begin{cases} \sum_{s=1}^{t_{max}^{S_i}} (d^s_{k\in S_i}) + \sum_{s=t^*}^{C_k} (d^s_{k\in S_i}) + t^* - t_{max}^{S_i} - 1 \\ \qquad\qquad if\ |\bigcap_{j\in S_i} H_j| < N\ \forall j \in S_i \\ \sum_{s=1}^{C_k} (d^s_{k\in S_i}) \quad if\ |\bigcap_{j\in S_i} H_j| = N\ \forall j \in S_i \end{cases} \tag{2}$$

*where $t_{max}^{S_i}$ is the last intra-cluster recovery stage for $S_i$ in the case of the non-availability of the $N$ packets at cluster members and $t^*$ is the first recovery stage of the inter-cluster recovery phase.*

*Proof:* To demonstrate the expression of the overall decoding delay of each cluster member $k \in S_i$ throughout the entire scenario, two cases are analyzed in terms of decoding delay: (i) all the $N$ packets are available in the cluster, (ii) not all the $N$ packets are available in the cluster. The complete proof is provided in Appendix A. □

*Corollary 2: For each cluster member $k \in S_i$, the individual completion time experienced throughout both recovery phases can be approximated as follows:*

$$C_{k\in S_i} = \frac{|W_k| + D_{k\in S_i} - q_k}{1 - q_k} \tag{3}$$

*where $|W_k|$ is the size of the WANTS set of $k$ and $q_k$ is the packet erasure probability, which is the average packet erasure probability linking $k$ to all remaining cluster members.*

*Proof:* The proof of this corollary is inspired by the work in [12] that considers a centralized scheme where the BS is the only transmitter of the decoding packets over one single recovery phase for all users. However, in our work since we consider a clustered CDE, multiple devices transmit to each other inside their clusters over both phases. The complete proof is provided in Appendix B. □

### C. CODED PACKET FORMATION AND IDNC GRAPH OVERVIEW

The problem of finding the optimal coding packet was examined in plenty of recent works to optimize IDNC performance metrics [12]–[16]. In our model, we use the packet combination technique that optimizes the completion time through decoding delay control [12]. In fact, since this problem is proven to be NP-hard, they propose a heuristic algorithm that minimizes the probability of increasing the completion time through a layered control of the decoding delay of each transmission. Therefore, the problem is shown to be equivalent to a maximum weight clique problem in which they designed a multi-layered IDNC graph [11] where each layer contains vertices that generate the bigger decoding delay values than those generated in the next layer and so on. The solution is the maximum clique composed of a number of vertices from where the suitable packets are extracted, combined and then transmitted. Let us discover the IDNC graph: To construct an IDNC graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a vertex $v_{ij} \in \mathcal{V}$ is created for every receiver $i$ missing packet $j \in W_i$. Two vertices $v_{ij}$ and $v_{lm}$ are connected with an edge $e \in \mathcal{E}$ if one of the two following conditions is verified:

- The receivers $i$ and $l$ miss the same packet, i.e. $j = m$
- The coded packet $j \oplus m$ is instantly decodable by both receivers $i$ and $l$, that is, $j \in H_l$ and $m \in H_i$

In order to efficiently reduce the complexity, and by the way the completion time, at every recovery transmission only the $\mathcal{CH}$, in a partition $\Pi = \{S_1, S_2, \ldots, S_m\}$, construct a local IDNC graph for their cluster members in order to determine the candidate network codes. Consequently, the graph $\mathcal{G}$ is a set of disjoint local graphs as follows:

$$\mathcal{G} = \begin{cases} \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\} \text{ if the second recovery phase} \\ \qquad\qquad \text{is not required} \\ \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m, \mathcal{G}_*\} \text{ if it exists a second recovery} \\ \qquad\qquad \text{phase in which } \mathcal{CH} \text{ construct } \mathcal{G}_* \end{cases}$$

### D. ENERGY CONSUMPTION MODEL

We consider that each node $k$ has a battery with a residual energy of $Es_k$, $k \in \mathcal{M}$. The simple energy model, that we have used in this paper, is introduced in [23]. It considers the inter-node distance $d$ and the free space $\epsilon_f$ ($d^2$ power loss) or multi path fading $\epsilon_m$ ($d^4$ power loss) channel model. Hence, the required energy for node $k$ to send an $L-$bit coded packet using the electrical energy $Eelec$ per bit and the threshold distance $d_{th}$ is:

$$Ec_k = \begin{cases} L \times Eelec + L \times \epsilon_f d^2 & \text{if } d \le d_{th} \\ L \times Eelec + L \times \epsilon_m d^4 & \text{if } d > d_{th} \end{cases} \tag{4}$$

## IV. COALITIONAL GAME IN PARTITION FORM FOR IDNC-BASED CDE NETWORK

Our main goal is to provide a distributed framework that can model the collaborations among the wireless nodes of an IDNC-based CDE network. To this end, we use the analytical framework of Cooperative Game Theory (CGT) [24], [25], which involves a set of players that interact with each other to form a partition. Particularly, in our work, we model the CDE as a coalition formation game in partition form with non-transferable utility, taking into account the energy efficiency and the completion time.

*Definition 4: A coalitional game in partition form with non-transferable utility (NTU) is defined by a pair* $(\mathcal{M}, \psi)$*, where* $\mathcal{M}$ *is the set of players and* $\psi$ *is a mapping such that for every partition* $\Pi$*, and coalition* $S \subseteq \mathcal{M}$, $S \in \Pi$, $\psi(S, \Pi)$ *is a subset of* $IR^{|S|}$ *representing the payoff vectors that players in* $S$ *can receive when cooperating (*$|S|$ *is the number of players in coalition* $S$*).*

The value of each coalition $S$ is a set of $|S|$ utilities, each of which is a function of the range of this coalition, the availability of the packets at members and also the availability of their stored energy. Thus, we propose a value function that takes into account the key metrics as follows:

Given coalition $S \in \Pi$, we define the coalition value set (obtained by all the coalitions) at stage $t \geq 1$ by:

$$\psi(S, \Pi)(t) = \{\psi_k(S, \Pi)(t) = -\alpha \times \frac{Ec_k(t)}{Es_k(t-1)} - T_S, \forall k \in S\} \quad (5)$$

where $\psi(S, \Pi)$ is a $|S|$-dimensional real vector whose element $k$ represents the utility that player $k$ can obtain within coalition $S$ in partition $\Pi$, $T_S = \|C_k(t)\|_\infty + \|D_k(t) - D_k(t-1)\|_1 \; \forall \; k \in S$, and $\alpha$ is a coefficient that tunes the weight of the energy consumption in the decision-making.

As one can clearly see, the utility of node $k \in S$ in (5) includes two main parts: energy consumption and cooperative delay. Both of them indicate the gain of forming coalitions. Indeed, several studies have considered a utility function as a combination of heterogeneous term, such as energy and throughput in [26] and energy and delay in [27]. The first term of the proposed utility function, which is the expected energy efficiency $\frac{Ec_k(t)}{Es_k(t-1)} \in [0, 1]$, captures the impact of the consumed energy when transmitting the recovery packet by player $k$ at stage $t$. $Ec_k(t)$ is the energy required to broadcast the recovery packet and $Es_k(t-1)$ is the stored energy of $k$ in the previous stage. Note that the energy cost increase when the residual energy of the node decreases. On the other hand, $T_S$ is the cooperative delay that takes into account the cluster completion time and the increase of the sum decoding delay over all players in the cluster between two consecutive stages. Moreover, in order to minimize both parameters simultaneously, there is a need to weight the terms of the value function. Therefore, we multiply the energy consumption part by a coefficient $\alpha$ that tunes the weight of the energy consumption in the decision-making.

Note that this utility function is used by $\mathcal{CH}$ in order to select the suitable decoding packet as well as the sender node. In fact, every $\mathcal{CH}$ chooses the best coalition member that sends the decoding packet with less consumed energy and targeting the maximum decoding nodes in the cluster.

*Proposition 1: The proposed clustered IDNC-based CDE is formulated as an* $(\mathcal{M}, \psi)$ *coalitional game in partition form with non-transferable utility.*

*Proof:* As given in definition 4, the coalition value in an NTU game is a set of payoff vectors. In our game, $\psi(S, \Pi)$ in (5) is a set of utility vectors since the term of the expected energy efficiency is related to each player in the coalition at every packet recovery transmission. Thus, it can be deduced that the proposed game is with NTU. On the other hand, if some packets are not existing in such a cluster $S$, the cluster head $CH_{k \in \{1, \cdots, |S|\}}$ cooperates with the other $CH_{l \in \{1, \cdots, |S|\} \backslash k}$ to recover the remaining missing packets in the inter-cluster recovery phase. Thus, $T_S$, and consequently $\psi(S, \Pi)$, does not only depend on players inside $S$, but also on players outside $S$ ie. $\Pi \backslash S$. Hence, from definition 4, we conclude that the proposed coalitional game is in partition form with NTU. ☐

## V. PROPOSED COALITION FORMATION ALGORITHM

According to the considered coalitional game model, we propose a merge-and-split algorithm to ensure the formation of the appropriate coalitional structure based on the nodes' preferences. Note that the proposed decision-making, i.e. two coalitions are merged or one is split is based on a preference order (or comparison relation) [28]. We define the collection of coalitions notion and the preference order in what follows.

*Definition 5: A collection of coalitions of* $\mathcal{M}$ *denoted* $S = \{S_1 \dots S_k\}$ *is a set of subsets of* $\mathcal{M}$*, not necessarily involving all players of* $\mathcal{M}$*. If a collection involves all players, it is called a partition of coalitions.*

*Definition 6: Given a partition* $\Pi$ *consisting of a set of coalitions in* $\mathcal{M}$*, a preference order* $\triangleright$ *is defined as a monotonic, transitive and irreflexive binary relation that compares any two coalitions of nodes* $S$ *and* $T \in \Pi$ *by comparing their utilities.*

Generally, as stated in [25], there exist two categories of preference orders; coalition-value orders and individual-value orders. The former compares two coalitions using their value (which is a single real number). Indeed, this category is suitable for TU-games. The latter compares two coalitions using their individual players payoffs. Since we have characterized our CDE game as an NTU-game in partition form, we choose an individual-value order called Pareto order $\triangleright$, which is adequate for NTU-games. This order is used only to compare partitions of the same set of players.

Let us consider two partitions of the set $\{s_1, s_2, \cdots, s_r\} \subset \mathcal{M}$, denoted by $P_1 = \{C_1, C_2, \cdots, C_k\}$ and $P_2 = \{C'_1, C'_2, \cdots, C'_l\}$. Consider two different partitions of $\mathcal{M}$ : $\Pi_1 = \{P_1, S_1, S_2 \dots S_n\}$ and $\Pi_2 = \{P_2, S_1, S_2 \dots S_n\}$ where $\{S_1, S_2 \dots S_n\}$ is a collection of $\mathcal{M}$.

We say that $P_1$ is preferred over $P_2$ by pareto order if and only if the following equation is satisfied:

$$(P_1, \Pi_1) \triangleright (P_2, \Pi_2) \Leftrightarrow \psi(s_i, \Pi_1) \geq \psi(s_i, \Pi_2), \quad \forall i \in \{1, \cdots, n\} \quad (6)$$

where there exists at least one node $s_j$ such that: $\psi(s_j, \Pi_1) > \psi(s_j, \Pi_2)$.

$\psi(s_i, \Pi_1)$ is the utility of the node $s_i$ when cooperating in partition $\Pi_1$ and $\psi(s_i, \Pi_2)$ is the utility of the node $s_i$ when cooperating in $\Pi_2$. Note that this preference order can also compare two coalitions in the same partition as well as two different coalitions in two different partitions.

In order to allow the nodes to build their suitable new structure based on the proposed preference order, we define the two following rules:

*Definition 7 (Split Rule):* In a given partition $\Pi_1$, a coalition $\bigcup_{i=1}^{l} S_i$ decides to split when $(\{S_1, \cdots, S_l\}, \Pi_1) \triangleright (\bigcup_{i=1}^{l} S_i, \Pi_2)$. Thus $\bigcup_{i=1}^{l} S_i \rightarrow \{S_1, \cdots, S_l\}$ and $\Pi_1 \rightarrow \Pi_2$ where $\Pi_2$ is the new formed partition after the operation of split.

*Definition 8 (Merge Rule):* In a given partition $\Pi_1$, the set of coalitions $\{S_1, \cdots, S_l\}$ decides to merge when $(\bigcup_{i=1}^{l} S_i, \Pi_2) \triangleright (\{S_1, \cdots, S_l\}, \Pi_1)$. Thus $\{S_1, \cdots, S_l\} \rightarrow \{\bigcup_{i=1}^{l} S_i\}$ and $\Pi_1 \rightarrow \Pi_2$ where $\Pi_2$ is the new formed partition after the operation of merge.

According to the preference order, only the $\mathcal{CH}$ make the merge and split decisions. Moreover, in order to reduce the complexity of the proposed algorithm, the split as well as the merge investigations are limited to dividing the coalition into two coalitions or merging two coalitions. Consequently, a coalition of players $S_i \in \Pi_1$ can be split, forming a new partition $\Pi_2$ where at least one node can enhance strictly its utility without hurting the payoffs of all remaining nodes in the new structure. Similarly, the decision of merging two disjoint coalitions $S_j$ and $S_i$ is assigned to both cluster heads $CH_j$ and $CH_i$.

*Remark 1*: In the proposed coalition formation algorithm, it is worth noting that the split and merge investigations depend on the payoffs of all players in the partition, due to the dependence of the game on externalities (partition form game).

In the initial phase, all players broadcast their feedback matrix allowing $\mathcal{CH}$ performing their first split iteration. Subsequently, merge operation begins. In fact, every $CH_i$ investigates all merge possibilities seeking the best coalition for merging. This candidate is determined in such a way that merge process improve both: cooperative delay and consumed energy of at least one player without harming any individual payoff. We assume that any $CH_i$ can start the merge process. The objective of the $\mathcal{CH}$ is to find a coalition structure that guarantees the lowest energy consumption and delay through a repetitive application of the above rules. Hence, when no further split nor merge operations happens, a new final partition is created in which all nodes will perform their clustered IDNC recovery phases. More details about our proposed algorithm are provided in Algorithm 1.

**Algorithm 1** Coalition Formation Algorithm for CDE

```
A-Initial phase
We start with a random partition $\Pi_1$ of $\mathcal{M}$ denoted by $\{S_1, S_2 \cdots, S_m\}$.
B-Split and merge phase
repeat
    a) Based on pareto order in (6), $\mathcal{CH}$ check the split action:
           $\Pi_2 = \textbf{\textit{Split}}(\Pi_1)$
       We obtain a novel partition $\Pi_2 = \{S_1, \ldots, S_p\}$
    b) for all $CH_i, i \in \{1, \cdots, p\}$ do
        1. $TO\_MERGE\_LIST_i = \{\}$
        2. $CH_i$ looks for coalition candidates $j$ for performing merge process and add them to its TO_MERGE list, each of which with its corresponding gain $G_{\{i,j\}}$:
               $TO\_MERGE\_LIST_i = \textbf{\textit{Examine}}(\Pi_2)$
       while $TO\_MERGE\_LIST_i$ is non-empty do
               $j^* = \textbf{\textit{argmax}}_{j \in \Pi_2 \setminus \{i\}} \{G_{\{i,j\}}\}$
        1. $CH_i$ sends REQ_TO_JOIN to $CH_{j^*}$;
               $\Pi_1 = \textbf{\textit{Merge}}(\Pi_2)$
       end while
       end for
until no successive merge and split operations occurs.
C-Cooperative data exchange recovery phase
All formed clusters in the final formed partition perform their intra-cluster recovery phases simultaneously and the inter-cluster recovery phase if necessary as described in Section III-A.
```

Let us introduce the defection function: the defection function notion (denoted *ID*), which consists in the association of a family of collection of partitions in $\mathcal{M}$, where each partition $\Pi$ involves two important stability forms: $ID_{hp}$ stability and $ID_c$ stability, each of which refers to a special defection function, respectively $ID_{hp}$ defection function and $ID_c$ defection function. First, as a result from [29], a partition $\Pi$ is $ID_{hp}$ stable when the coalition members have no incentive to execute further merge nor split operations. However, a partition $\Pi$ is $ID_c$ stable when no node has the incentive to move from that partition and form any other new collection in $\mathcal{M}$. After executing the proposed algorithm (Algorithm 1), nodes are arranged into independent disjoint coalitions in a novel partition.

*Theorem 1: Any network partition resulting from the proposed merge and split algorithm is $ID_{hp}$ stable.*

*Proof:* In our merge and split algorithm, we are using the Pareto order to merge or split two coalitions. Hence, after the merge or the split operation, the utility of the nodes in the target coalitions is higher or equal to their utility in the current configuration (at least one node should strictly increase its utility without harming other nodes). Hence, successive merge and split iterations produce a sequence of partitions $\Pi_1, \Pi_2, \cdots$ with $\Pi_{i+1} \triangleright \Pi_i \ \forall i \geq 1$. However, the number of different partitions of a finite set of node is finite. Therefore, by transitivity and irreflexivity of Pareto order, a partition $\Pi$ cannot be revisited by the merge and split algorithm and the sequence of merge and split is finite. Thus, the termination of the two rules iterations is guaranteed and then we conclude that the proposed merge-and-split algorithm converges to a final partition $\Pi_{fin}$. Suppose that this final resulting partition $\Pi_{fin} = \{S_1, \cdots, S_l\}$ is not $ID_{hp}$ stable. Then there exists two coalitions $S_i, S_j \in \Pi$ that are interested to perform a merge, i.e. $(S_j \cup S_i, \Pi'_{fin}) \triangleright (\{S_j, S_i\}, \Pi_{fin})$ with $i \neq j$ or a coalition $S_i \in \Pi$ interested in splitting over two coalitions $S_i = S_i^1 \cup S_i^2$, i.e. $(\{S_i^1, S_i^2\}, \Pi'_{fin}) \triangleright (S_i, \Pi_{fin})$. Hence, there exists a new partition $\Pi'_{fin}$ resulting from merge or split operations such that $\Pi'_{fin} \triangleright \Pi_{fin}$, which leads to a contradiction since $\Pi_{fin}$ is preferred over all the possible partitions obtained through merge and split operations. Thus, any obtained partition resulting from the proposed merge-and-split algorithm is $ID_{hp}$ stable. □

### A. COMPLEXITY ANALYSIS

The merge-and-split algorithm has a complexity far lower than the coalition formation problem in optimal manner which is NP-hard [30], [31]. In fact, we have to check all the possible partitions, which is equal to the $M$-th Bell number $B_M$, in order to find the optimal partition. Note that the Bell number is obtained by the recursion $B_{n+1} = \sum_{k=0}^{n} \binom{k}{n} B_n$, $B_0 = B_1 = 1$. In this regard, among the diverse algorithmic solutions, we have chosen the merge and split allowing the partition of nodes in a distributed fashion. Indeed, it is the most suitable algorithmic solution for our proposed game theoretic solution due to its low complexity, its adaptation possibility within partition form games and the distributed nature of the CDE problem.

The complexity of the proposed merge and split algorithm depends on the number of merge-and-split investigations performed in every iteration, which depends on the number of nodes in the network. In fact, each coalition needs to test the merge with all the other coalitions in $\Pi$ (worst case scenario). Hence, the total number of merge attempts is at most $O(|\Pi|^2)$, which depends on the number of coalitions and not on the number of nodes in the network. However, since the merge operation is executed by coalition heads $\mathcal{CH}$ in a distributed manner, the complexity of the merge operation for each coalition is $O(|\Pi|)$.

Regarding the splitting operation, the total number of attempts in the worst case implies finding all possible partitions of the coalition, which gives a worst case complexity for the coalition $S_k$ of $O(\left\{ {2 \atop |S_k|} \right\})$ where $\left\{ {2 \atop |S_k|} \right\}$ is the Stirling number of the second kind that counts the number of ways to divide the coalition $S_k$ into two new coalitions. Therefore, the complexity of the split operation is closely related to the size of the coalition and not on the total number of users in the system. On the other hand, as mentioned in section III.C, in our proposed scheme, only $\mathcal{CH}$ are in charge of executing the heuristic algorithm [22] for determining the suitable combined packets. Hence, the complexity of cheking the connectivity of each vertex with the other vertices and renewing its corresponding weight and layer is limited to the cluster size. It is equal to $O(|S_k|N)$ where $|S_k|$ is the size of a cluster $S_k$ and $N$ is the number of packets. The reason is that each vertex can be only connected to vertices in the same local graph $\mathcal{G}_k$ which is composed of at most $|S_k|N$ vertices. Therefore, the total complexity for all the network is $O(|\mathcal{CH}||S_k|N)$. However, for one grand coalition, only one big graph is constructed by every node in the network which consists at most of $MN$ vertices. Then, the total complexity for determining the suitable packet is $O(M^2N)$.

### B. MOBILE COOPERATIVE UAV NETWORK: A CASE STUDY

As a special form of mobile ad hoc network (MANET) and vehicular ad hoc network (VANET), the network of multi-UAV is classified as flying ad hoc network (FANET) [3]–[5] since it presents different characteristics such us node density, power consumption, computation power, frequency of topology changes and node mobility compared to other categories of ad hoc network. Let us focus on topology changes and node mobility. For example, consider a network of drones that are arranging into a random partition of collaborating coalitions. Applications like data-collection from the sink of a sensor network or monitoring an area do not require the drones fleet configuration to change.

However, applications like forest surveillance and monitoring require that some drones move across the target area. Consequently, the distribution of some coalitions may change in such a way its members could be scattered or the inter-drone distance may increase, and then the recovery phases execution could not be energy-efficient anymore. In such case, they have to execute the proposed coalition formation process to be able to re-arrange themselves into a novel energy-efficient stable partition. When starting, they may use two different configurations: (1) They act as a single grand coalition where it exists a cluster head that will start by the split investigation or (2) They keep their current clustered partition where every cluster head iteratively applies the merge and split rules. As small UAVs have limited computational capabilities [3], deciding how they can be arranged upon starting is important. In fact, the more there are drones per coalition, the higher is the number of split investigations and the higher the complexity of the algorithm is.

## VI. SIMULATION RESULTS

This section presents a comprehensive Matlab-based simulation of the proposed solution. Simulation results show the average of the cluster-completion time and the total consumed energy with a tuning energy coefficient $\alpha = 10$ of $M$ devices until recovering all $N$ packets of the frame over several runs. The packet size is 8 bytes. In each run, all wireless nodes are randomly located in a square field of $150 \times 150 \ m^2$ and the node-to-node packet erasure probability is changed for each new run. All simulation parameters are listed in Table 1.

**TABLE 1.** Simulation parameters.

| Parameter | Value |
|---|---|
| Area | $150 \times 150\, m^2$ |
| $\epsilon_f$ | $10\, pJ/bit/m^2$ |
| $\epsilon_m$ | $0.0013\, pJ/bit/m^4$ |
| $Eelec$ | $50\, nJ/bit$ |
| $d_{th}$ | 25 m |
| $\alpha$ | 10 |
| Packet size | 64 bits |

In particular, two applications are considered: a multi-drones network and a wireless sensor network, each of which is evaluated in a seperate part as follows.

In the first part, the performance of our proposed solution is compared against the two following IDNC-based schemes:

- 'Delay-aware and Energy-unaware non-coalitional CDE' which considers a non-cooperative game in a D2D configuration to select a single transmitting device among a number of players arranged in a single big cluster in order to reduce only the overall completion time [15].
- 'Delay and Energy-aware non-coalitional CDE' which uses the same model of the previous scheme but with a modified utility function. In fact, in this scheme, our proposed utility function is considered to select the transmitting device in order to reduce both the overall completion time and the total consumed energy.

Note that the total consumed energy consists of the consumed energy when sending the recovery packets in addition to the consumed energy when exchanging feedback messages after every reception of a decoding packet throughout all the scenarios. All results are presented while increasing the packet erasure probability $Q$ from the sender (the BS for example) to all wireless nodes during the initial phase. Moreover, we do the same analysis with respect to the number of devices while the packet erasure probability $Q$ remains constant.

Moreover, a scenario of topology change is considered in a drone network which is already partitioned, where a number of mobile drones move randomly from their localization. Thus, we analyze and compare the performance of both resulting partitions in the two following cases:

- when they start from the grand coalition,
- when they start from their current clustered partition.

Obviously, in each run, for the same distribution of nodes, we measure the parameters of the two aforementioned resulting partitions using different state matrices and then average them over all runs. In particular, in such analysis, not only the consumed communication energy is considered but also the total computation energy consumed by cluster heads when running the proposed merge-and-split algorithm is also taken into account. Thus, we use the computation energy model introduced in [32]. Drones are powered by Intel Atom x7-Z8700 processor [33]. Then, we calculated the number of instructions in our merge-and-split algorithm according to the intel Instruction Set Reference [34].

In the second part, we assess our proposed scheme in a WSN where the sensors are randomly clustered at the beginning in order to guarantee low complexity processing of the coalition formation algorithm.

### A. APPLICATION IN A DRONE FLEET NETWORK

In this section, we focus on a drone fleet that collects information from the sink of wireless sensor network. Since drones are of high mobility, we investigate the topology changes at the end of this section. Figures 3 and 4 depict respectively the average completion time per cluster and the consumed energy by all drones in the network depending on the erasure probability $Q$ for a scenario where $M = 10$ drones and $N = 20$ packets. From the aforementioned figures, it can be observed that the proposed cooperative framework provides a significant completion time and energy consumption reduction as compared to the two other non-coalitional schemes. Moreover, figure 5 illustrates the delay gain and energy-consumption gain per node when using our proposed cooperative scheme. It can be observed that gains on delay decrease as $Q$ increases. This can be explained by the fact that when $Q$ increases, the cardinality of the HAS set per drone decreases, and thus the probability that the union of the HAS sets of devices in a smaller coalition is equal to $\mathcal{N}$ is low. Hence, the inter-cluster recovery phase is always required for all the clusters which would slow down the recovery phase. Therefore, drones would have less incentive to cooperate. Furthermore, energy gains shown in figure 5 increase until $Q = 0.4$ and despite a very high packet erasure probability, the proposed coalition formation algorithm yields a performance improvement on energy consumption of 31.38% ($Q = 0.6$) against the Delay-aware and Energy-unaware non-coalitional CDE.

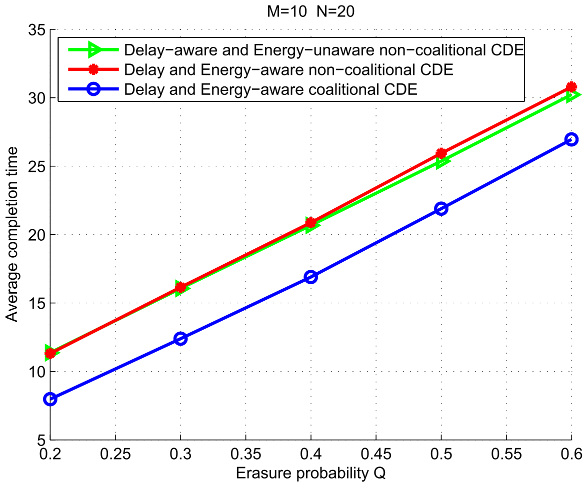

**FIGURE 3.** Average cluster-completion time of the resulting clustered network versus the two non-coalitional models with respect to packet erasure probability Q.

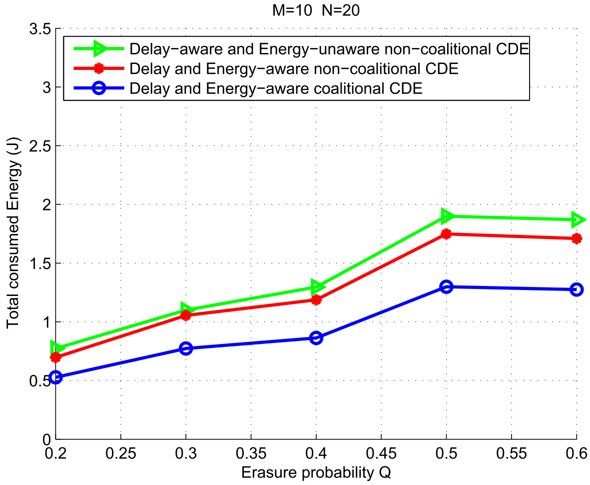

**FIGURE 4.** **Total energy consumption in the resulting clustered network versus the two non-coalitional models with respect to packet erasure probability Q.**

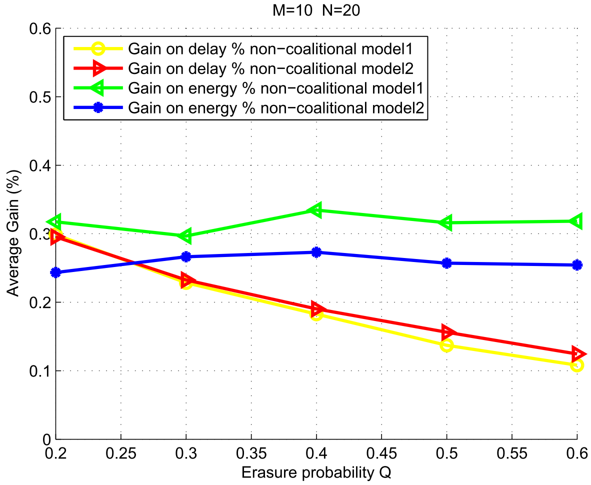

**FIGURE 5.** **Average gains per node achieved by the resulting clustered network with respect to packet erasure probability Q. The non-coalitional model1 is delay-aware and energy-unaware non-coalitional CDE and the non-coalitional model2 is delay-aware and energy aware non-coalitional CDE.**

Figures 7 and 8 depict respectively the total energy consumption in the network and the average completion time per cluster as the drone fleet size increases for a scenario of $Q = 0.2, N = 20$ packets. It can be observed that the proposed scheme outperforms the Delay-aware and Energy-unaware non-coalitional CDE in terms of both: completion time and energy consumption. From figure 6, we notice that the benefit of cooperation in terms of energy and delay increases with the number of drones. In other words, the presence of more drones in the field enhances the incentive of cooperation. This is mainly due to two reasons: On one hand, small coalitions attempt simultanously to finish earlier their recovery phases compared to the one big coalition. On the other hand, exchanging recovery packets combinations and feedback matrices among a reduced number of drones is performed in a smaller range compared to the grand coalition.

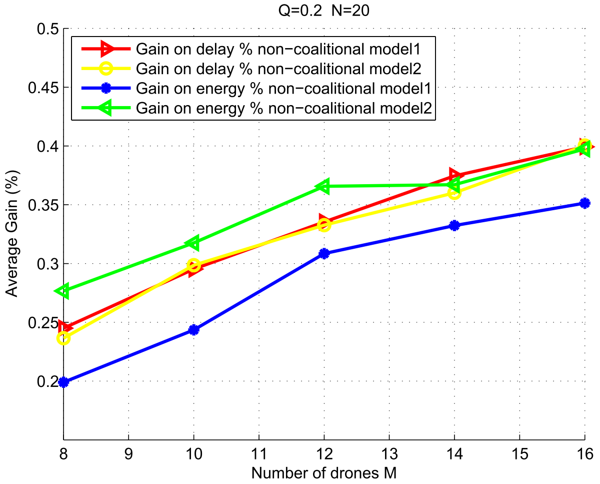

**FIGURE 6.** **Average gains per node achieved by the resulting clustered network with respect to drone fleet size. The non-coalitional model1 is delay-aware and energy-unaware non-coalitional CDE and The non-coalitional model2 is delay-aware and energy aware non-coalitional CDE.**

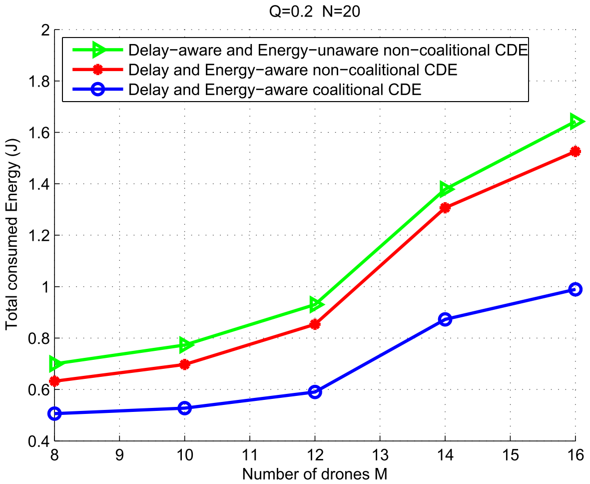

**FIGURE 7.** **Total energy consumption of the drones in the resulting clustered network versus the two non-coalitional models with respect to drone fleet size.**

All these figures demonstrate the significant advantage of using our clustered CDE scheme in terms of both delay and energy, which is increasing with drone fleet size reaching up to 39.75% of improvement in energy consumption and 40% of improvement in completion time compared to the non-coalitional model of the Delay-aware and Energy-unaware non-coalitional CDE when $M = 16$ drones.

### 1) TOPOLOGY CHANGES EVALUATION

After an environmental change, the objective is to investigate the adequate starting partition that allows drones to converge to a novel partition where they can reduce not only the communication energy and delay but also the computation energy consumed when running the coalition formation process. In figure 9, we can clearly observe that at any size of the fleet,

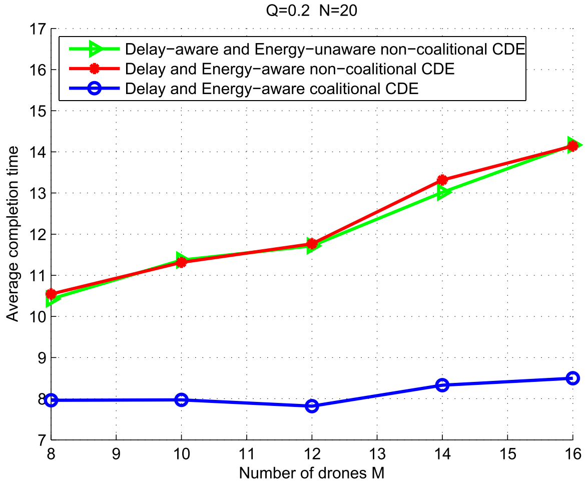

**FIGURE 8.** **Average cluster-completion time achieved by the resulting clustered network versus the two non-coalitional models with respect to drone fleet size.**

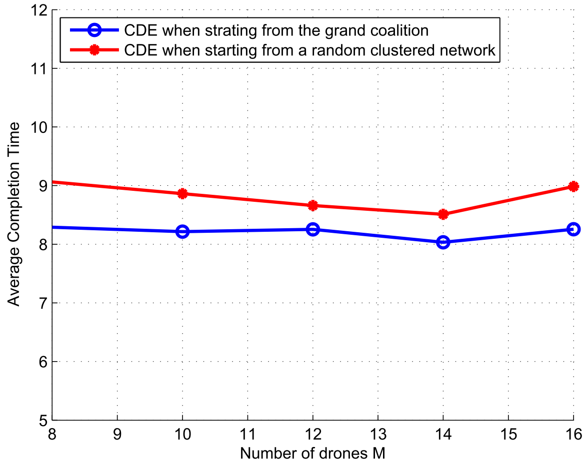

**FIGURE 9.** **Average cluster-completion time achieved by the resulting clustered network when starting by the grand coalition versus the resulting clustered network when starting by a random clustered partition with respect to drone fleet size.**

when drones process the coalition formation phase as a single grand coalition, they completely decode their missed packets faster than starting with a random clustered structure. As we have detailed in section 5.1, this result is expected since the number of possible partitions that are examined by cluster heads throughout all the phase is significantly high. Hence, finding the lower completion time among all those possibilities is guaranteed. On the other hand, figure 10 depicts the total consumed energy taking into account the computation energy of both resulting partitions with respect to the number of drones. It is particularly interesting to observe that when the size of the fleet is less than 12 drones, starting as a single grand coalition allows drones in the resulting structure to further reduce their energy consumption. However, once exceeding 12 drones, starting with grand coalition is not the optimal choice anymore. In fact, it requires processing a very high number of split attempts, then processing a very high number of instructions that causes a substancial increase of the computation energy compared to the clustered starting partition as it is illustrated in the computation energy curves in figure 10.

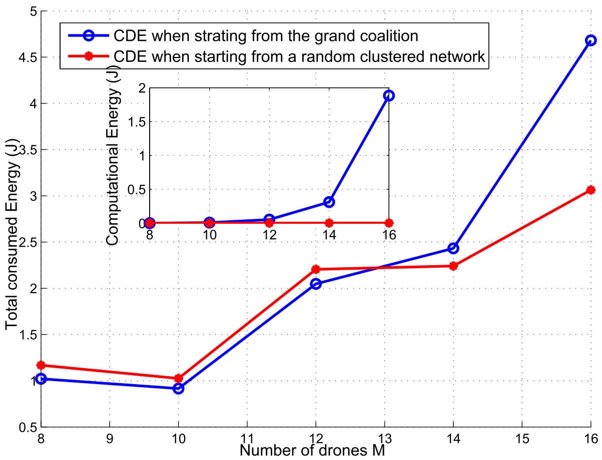

**FIGURE 10.** **Total energy consumption taking into account the computational energy achieved by the resulting clustered network when starting by the grand coalition versus the resulting clustered network when starting by a random clustered partition with respect to drone fleet size.**

### B. APPLICATION IN A WIRELESS SENSOR NETWORK

We consider, in this section, a WSN where nodes are interested in receiving the same set of packets.

Figures 12 and 13 depict respectively the total energy consumption in the network and the average completion time per cluster as the number of sensors increases, for a scenario of $N = 20$ when the packet erasure probability $Q = 0.2$. Figure 12 illustrates that the benefit of using our cooperative scheme is increasing with the number of users. We can clearly observe that the gap between the total consumed energy of our proposed coalition formation algorithm and the total consumed energy of the starting partition is increasing as $M$ increases, reaching up 33.55% of improvement of energy consumption when we have 60 cooperating sensors. In fact, the more we have sensors in the field, the more they have an incentive to construct more clusters in order to exchange the recovery packets combination as well as the feedback matrices among a reduced number of sensors in a smaller range. In figure 11, we present an example of a simulated scenario consisting of $M = 30$ sensor nodes. At the beginning, sensors are arranged into three large coalitions. Therefore, after the execution of our proposed algorithm, a final resulting $ID_{hp}$-stable network partition is generated. As we can Cleary see, it consists of ten disjoint smaller coalitions each of which is composed at least of two sensors. On the other hand, figure 13 illustrates the significant improvement of the completion time in the new structure reaching 29.8% when $M = 60$ sensors. In fact, the presence of more sensors in the field enhances the incentive to form more cooperating coalitions number attempting to finish earlier their recovery phase.

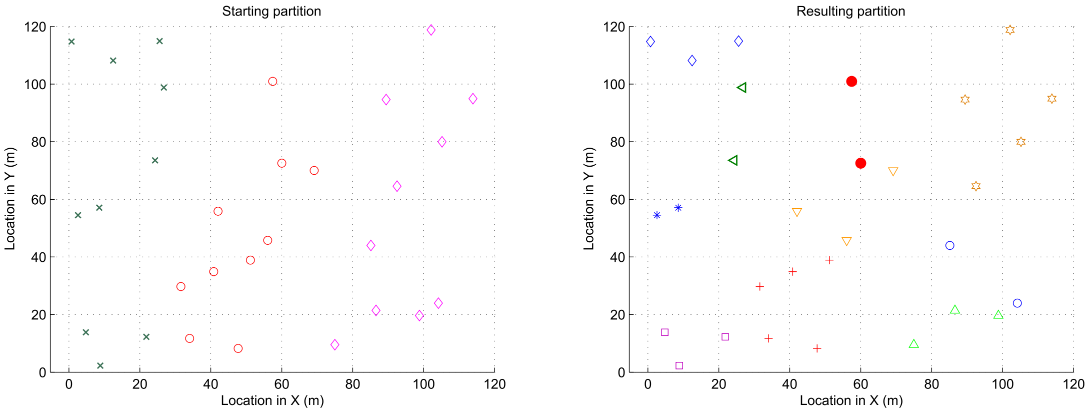

**FIGURE 11.** Convergence of the algorithm to a final $ID_{hp}$-stable partition.

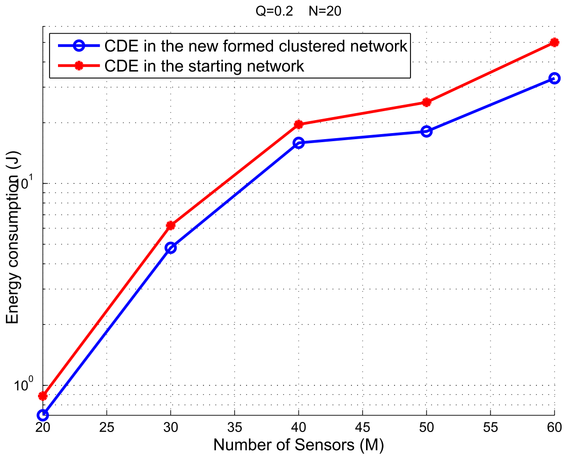

**FIGURE 12.** Total energy consumption of the sensors in the resulting clustered network versus the starting partition with respect to the number of sensors.

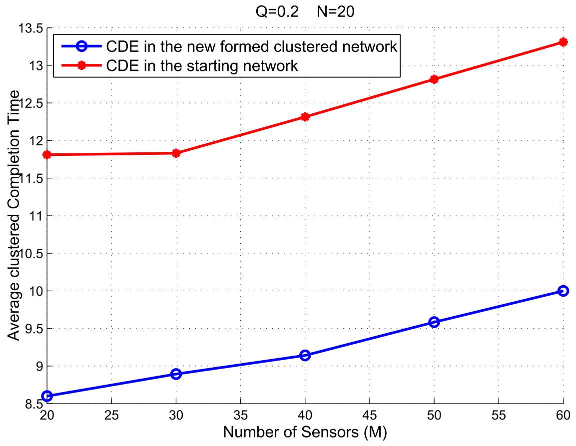

**FIGURE 13.** Average cluster-completion time achieved by the resulting clustered network versus the starting partition with respect to the number of sensors.

## VII. CONCLUSION

In this paper, we have studied the problem of joint-minimization of completion time and energy consumption in the cooperative data exchange using the instantly decodable network coding across wireless nodes having a limited battery capacity. We modeled the problem using cooperative game theory in partition form in which the players seek to form a disjoint coalitions that reduce both the completion time and energy consumption. To solve the game, we have proposed a distributed merge and split algorithm that is guaranteed to converge to a stable network. Moreover, we addressed the mobility issue through multi-UAVs network. Simulation results have shown that our proposed cooperative game theoretical framework, by considering an additional constraint that is the energy consumption, reduces both average completion time and energy consumption for the resulting clustered network. Note also that using the coalitional game theoretic framework enhances the scalability of the system since each cluster head have to maintain a feedback matrix of the cluster's members instead of the global feedback matrix, like the non-cooperative model.

## APPENDIX A
## PROOF OF COROLLARY 1

Suppose we have a partition of coalitions composed of $n$ coalitions of collaborating nodes $S = \{S_1, S_2, ..S_n\}$. All clusters in the network are executing the clustered IDNC protocol to recover their missing packets. Let us consider a cluster $S_i$ composed of $m$ nodes and let $k$ be the $k^{th}$ node in $S_i$. To compute the overall decoding delay of $k$ throughout both recovery phases, we have to consider two cases:

1) All packets are available in $S_i \Leftrightarrow |\bigcap_{j=1\in S_i}^{m} H_j| = N$. In that case, nodes in $S_i$ do not need to wait for the inter-cluster phase to receive their remaining wanted packets. Note that node $k$ completes receiving all its erased packets in the $C_k$-th transmission. Therefore, according to definition 3, the overall decoding delay experienced by $k$ is simply expressed as follows: $D_{k\in S_i} = \sum_{s=1}^{C_k}(d^s_{k\in S_i})$
2) Not all packets are available in $S_i \Leftrightarrow |\bigcap_{j=1\in S_i}^{m} H_j| < N$ In that case, the overall decoding delay of node $k$ can be divided into three terms $D_1$, $D_2$ and $D_3$, each of

which expresses the effect of an occurring event on the decoding delay as follows:

- $D_1$ is the accumulative decoding delay experienced by $k$ during the intra-cluster recovery phase until all cluster members still miss only the unavailable packets (at $t = t_{max}^{S_i}$), ie. $\forall j \in S_i$, $W_j = W_{\tilde{j} \in S_i \setminus \{j\}}$. Therefore, $D_1 = \sum_{s=1}^{t_{max}^{S_i}} (d_{k \in S_i}^{s})$
- After completing the first recovery phase at $t = t_{max}^{S_i}$, nodes in $S_i$ should wait other clusters in the network completing their intra-cluster recovery processes. Since there is no decoding packets, the decoding delay of each device is increased by $D_2$ units. Therefore, $D_2 = (t_{max}^{S_j} + 1) - (t_{max}^{S_i} + 1)$ where $S_j$ is the last finishing cluster. Note by $t^*$ the first recovery stage of the second phase, then $D_2 = t^* - t_{max}^{S_i} - 1$.
- $D_3$ is the decoding delay experienced by the cluster head $CH_i$ in the inter-cluster recovery phase. In fact, if the received packet is instantly decodable, it will be forwarded by $CH_i$ to its cluster members, otherwise, no packet is forwarded. Therefore, in that case, the decoding delay of device $k$ is the same as the cluster head: $D_3 = \sum_{s=t^*}^{C_k} (d_{k \in S_i}^{s})$.

## APPENDIX B
## PROOF OF COROLLARY 2

Let us first examine all possible packet transmissions closely among cluster members that may affect the individual completion time throughout both recovery phases. Let $\mathrm{F_k(t)}$ be the total number of erased coded packets at receiver $k$ until time slot $t$. Note that a device $k$ receives its last instantly decodable packet at time $t = C_k$. Thus, until $t = C_k - 1$, one of the following cases may happen:

- The coded packet is erased, thus $\mathrm{F_k}(t) = \mathrm{F_k}(t-1) + 1$.
- The coded packet is successfully received. Two cases are possible:
  - The combination of packets is instantly decodable for device $k$ so it needs $|W_k(0)| - 1$ such coded packets to recover all the remaining missing ones.
  - The combination of packets is non-innovative or not instantly decodable for device $k$. Thus, its accumulative decoding delay $D_{k \in S_i}^{t}$ at $t <= C_k - 1$ increases by one unit.
- No coded packet is received. One of the two following reasons can be considered for this case:
  - The non-availability of a number of packets at any member of the cluster $S_i$ of device $k$. In that case, the decoding delay of $k$ is increasing by one unit at every stage until the beginning of the inter-cluster recovery phase when all remaining clusters finish their intra-cluster recovery exchanges as detailed in Corollary 1.
  - The reception of the cluster head of a coded packet (from another cooperating cluster head) which is non-innovative or non instantly decodable in the second phase. Therefore, there is no relayed decoded packet for its cluster members.

Consequently, the number of required recovery transmission $C_k$ until device $k$ belonging to cluster $S_i$ receives all its wanted packets can be expressed as follows:

$$C_{k \in S_i} = |W_k(0)| + D_k^{C_k} + F_k(C_k - 1) \tag{B.1}$$

Since the $C_k$-th transmission is the last successful transmission that allows node $k$ to complete the reception of lost packets, $F_k(C_k - 1) = F_k(C_k)$, therefore:

$$C_{k \in S_i} = |W_k(0)| + D_k^{C_k} + F_k(C_k) \tag{B.2}$$

Let $\mathcal{Y}_k(t)$ be a bernoulli random variable that is equal to 0 if the packet is successfully received at time $t$ and 1 if it is erased:

$$\mathbb{P}(\mathcal{Y}_k(t) = y) = \begin{cases} q_k & \text{if } y = 1 \\ 1 - q_k & \text{if } y = 0 \end{cases} \tag{B.3}$$

Let $\mathcal{J}(t)$ be a random variable taking the chosen sender index $k'$ within the cluster $S_i$. The probability of packet erasure at device $k$ in the transmission $t$ is calculated as:

$$\mathbb{P}(\mathcal{Y}_k(t) = 1) = \sum_{k' \in S_i} \mathbb{P}(\mathcal{Y}_k(t) = 1 | \mathcal{J}(t) = k') \mathbb{P}(\mathcal{J}(t) = k') \tag{B.4}$$

Note that if the sender is itself the receiver ie. $k' = k$, the coded packet cannot be erased, thus $\mathbb{P}(\mathcal{Y}_k(t) = 1 | \mathcal{J}(t) = k) = 0$ otherwise (if $k \neq k'$) and according to the system model, the erasure probability between two nodes $k$ and $k'$ is equal to:

$$\mathbb{P}(\mathcal{Y}_k(t) = 1 | \mathcal{J}(t) = k') = q_{k'k} \tag{B.5}$$

On the other hand, since all devices start with the same residual energy supply, all devices have the same chance to be selected as a sender in its cluster. Hence:

$$\mathbb{P}(\mathcal{J}(t) = k') = \frac{1}{|S_i|}, \quad \forall k' \in S_i \tag{B.6}$$

Replacing B.5 and B.6 in B.4, the probability that the coded packet is erased at device $k$ is expressed as follows:

$$\mathbb{P}(\mathcal{Y}_k(t) = 1) = \frac{1}{|S_i|} \sum_{k' \neq k \in S_i} q_{k'k} = \frac{|S_i| - 1}{|S_i|} \bar{q_k} \tag{B.7}$$

where $\bar{q_k} = \frac{1}{|S_i| - 1} \sum_{k' \neq k \in S_i} q_{k'k}$ is the average packet erasure probability of device k in the cluster $S_i$. Hence, the cumulative number of erased packets at device $k$ until $t = C_k - 1$ is the sum of $C_k - 1$ bernoulli variable as follows:

$$F_k(C_k - 1) = \sum_{t=1}^{C_k - 1} \mathcal{Y}_k(t) \tag{B.8}$$

For a large number of packets, the individual completion time $C_k$ would be automatically large. Using the law of numbers, $F_k(C_k - 1)$ is approximated as follows:

$$F_k(C_k - 1) = (C_k - 1) \frac{|S_i| - 1}{|S_i|} \bar{q_k} \tag{B.9}$$

After substituting B.9 into the completion time expression B.1, the individual completion time for device $k$ can be finally calculated as follows:

$$C_{k\in S_i} = \frac{|W_k(0)| + D_{k\in S_i} - \frac{|S_i|-1}{|S_i|}\bar{q}_k}{1 - \frac{|S_i|-1}{|S_i|}\bar{q}_k} \tag{B.10}$$

## REFERENCES

[1] A. Nordrum, "The Internet of fewer things [news]," *IEEE Spectr.*, vol. 53, no. 10, pp. 12–13, Oct. 2016.

[2] S. E. Tajbakhsh and P. Sadeghi, "Coded cooperative data exchange for multiple unicasts," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Lausanne, Switzerland, Sep. 2012, pp. 587–591.

[3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[4] J.-J. Wang, C.-X. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[5] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[8] J. K. Sundararajan, D. Shah, and M. Medard, "Online network coding for optimal throughput and delay—The three-receiver case," in *Proc. IEEE Int. Symp. Inf. Theory Appl. (ISITA)*, Auckland, New Zealand, Dec. 2008, pp. 1–6.

[9] A. Douik, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Instantly decodable network coding: From centralized to device-to-device communications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1201–1224, 2nd Quart., 2017.

[10] A. Sprintson, "Cooperative data exchange in broadcast networks," in *Cooperative and Graph Signal Processing*. New York, NY, USA: Academic, 2018, pp. 489–504.

[11] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1553–1567, Oct. 2015.

[12] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Decoding-delay-controlled completion time reduction in instantly decodable network coding," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2756–2770, Mar. 2017.

[13] M. S. Karim, A. Douik, and S. Sorour, "Rate-aware network codes for video distortion reduction in point-to-multipoint networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7446–7460, Aug. 2017.

[14] Y. Keshtkarjahromi, H. Seferoglu, R. Ansari, and A. Khokhar, "Content-aware network coding over device-to-device networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2147–2158, Aug. 2017.

[15] A. Douik, S. Sorour, H. Tembine, T. Y. Al-Naffouri, and M.-S. Alouini, "A game-theoretic framework for network coding based device-to-device communications," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 901–917, Apr. 2017.

[16] A. Douik and S. Sorour, "Data dissemination using instantly decodable binary codes in fog-radio access networks," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2052–2064, May 2018.

[17] M. S. Karim, A. Douik, P. Sadeghi, and S. Sorour, "On using dual interfaces with network coding for delivery delay reduction," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3981–3995, Jun. 2017.

[18] W. C. Y. Lee, *Mobile Communication Design Fundamentals*, 2nd ed. Hoboken, NJ, USA: Wiley, 2010.

[19] J. S. Leu, T. H. Chiang, M. C. Yu, and K. W. Su, "Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes," *IEEE Commun. Lett.*, vol. 19, no. 2, pp. 259–262, Feb. 2015.

[20] F. U. Khan, I. A. Shah, S. Jan, I. Khan, and M. A. Mehmood, "Fuzzy logic based cluster head selection for homogeneous wireless sensor networks," in *Proc. IEEE Int. Conf. Open Source Syst. Technol. (ICOSST)*, Lahore, Pakistan, Dec. 2015, pp. 41–45.

[21] S. Hadzic, J. Bastos, and J. Rodriguez, "Reference node selection for cooperative positioning using coalition formation games," in *Proc. 9th Workshop Positioning Navigat. Commun. (WPNC)*, Dresden, Germany, Mar. 2012, pp. 105–108.

[22] 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. IEEE Global Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 5008–5013.

[23] M. Tarhani, Y. S. Kavian, and S. Siavoshi, "SEECH: Scalable energy efficient clustering hierarchy protocol in wireless sensor networks," *IEEE Sensors J.*, vol. 14, no. 11, pp. 3944–3954, Nov. 2014.

[24] R. B. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard Univ. Press, Sep. 1991.

[25] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 77–97, May 2009.

[26] G. Yang, "A non-cooperative game approach for power control MAC in wireless sensor networks," *Tehnički Vjesnik*, vol. 22, no. 2, pp. 303–310, 2015.

[27] B. S. Jamin, M. Chatterjee, and T. Samanta, "A game theoretic routing framework based on energy-delay conservation in WSNs," in *Proc. IEEE Int. Conf. Intell. Sensor, Sensor Netw. Inf. Process.*, Apr. 2015, pp. 1–5.

[28] K. Apt and A. Witzel, "A generic approach to coalition formation," in *Proc. IEEE Workshop Comput. Social Choice (COMSOC)*, Amsterdam, The Netherlands, Dec. 2006, pp. 347–367.

[29] K. R. Apt and T. Radzik. (May 2006). "Stable partitions in coalitional games." [Online]. Available: https://arxiv.org/abs/cs/0605132

[30] W. Saad, Z. Han, M. Debbah, and A. Hjorungnes, "A distributed coalition formation framework for fair user cooperation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4580–4593, Sep. 2009.

[31] L. Militano *et al.*, "A constrained coalition formation game for multihop D2D content uploading," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2012–2024, Mar. 2016.

[32] A. Wang and A. Chandrakasan, "Energy-efficient DSPs for wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 4, pp. 68–78, Jul. 2002.

[33] Intel Atom. *Processor Z8000 Series Datasheet*. Accessed: Mar. 2016. [Online]. Available: http://www.intel.com/content/www/us/en/processors/atom/atom-z8000-datasheet-vol-1.html

[34] *Intel 64 and IA-32 Architectures Software Developer's Manual*, Intel Atom, Oct. 2016. [Online]. Available: https://software.intel.com/en-us/articles/intel-sdm

Authors' photographs and biographies not available at the time of publication.