IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 19, 2020, accepted August 26, 2020, date of publication August 31, 2020, date of current version September 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020472

Coalition Formation Game for Cooperative Content Delivery in Network Coding Assisted D2D Communications

MOHAMMED S. AL-ABIAD[1], (Student Member, IEEE), AHMED DOUIK[2], (Member, IEEE), AND MD. JAHANGIR HOSSAIN[1], (Senior Member, IEEE)

[1]School of Engineering, The University of British Columbia, Kelowna, BC V1V 1V7, Canada
[2]Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA

Corresponding author: Mohammed S. Al-Abiad (m.saif@alumni.ubc.ca)

This work was supported by the Discovery Grant of Natural Science and Engineering Council (NSERC) of Canada.

ABSTRACT Device-to-device (D2D) communications have shown a huge potential in cellular offloading and become a potential technology in 5G and beyond. In D2D networks, the requested contents by user devices (UDs) can be delivered via D2D links, thus offloading the content providers (CPs). In this work, we address the problem of minimizing the delay of delivering content in a *decentralized* and *partially* D2D connected network using network coding (NC) and cooperation among the UDs. The proposed optimization framework considers UDs' acquired and missing contents, their limited coverage zones, NC, and content's erasure probability. As such, the completion time for delivering all missing contents to all UDs is minimized. The problem is modeled as a coalition game with cooperative-players wherein the payoff function is derived so that increasing individual payoff results in the desired cooperative behavior. Given the intractability of the formulation, the coalition game is relaxed to a coalition formation game (CFG). A distributed coalition formation algorithm relying on merge-and-split rules is developed for solving the relaxed problem at each transmission. The effectiveness of the proposed solution is validated through computer simulation against existing schemes.

INDEX TERMS Coalition game, content delivery, device-to-device networks, instantly decodable network coding, multimedia streaming, real-time applications.

I. INTRODUCTION

A. BACKGROUND AND RELATED WORKS

The use of smartphones and data-hungry applications in radio access networks are increasing dramatically worldwide. This growth impacts the ability of traditional wireless networks to meet the required Quality-of-Service (QoS) for its user devices (UDs). Device-to-device (D2D) communication, as one of the candidate technologies for 5G and beyond [1], [2], supports a massive number of connected devices and possibly improves the data-rate without fixed infrastructure for content delivery [3], [4]. The decentralized nature of D2D networks allows devices to communicate with other nearby devices via D2D links, which is suitable for numerous applications in mobile networks. For example, in wireless cellular networks, a D2D system enables cellular offloading by cooperation among UDs for content downloading and sharing.

The report provided by Cisco [5] shows that most of the wireless data contents are mainly distributed in hotspot areas, such as a playground, public transport, a conference hall. In such hotspots, there is a tremendous high UD density and UDs request high volume of data traffic. As such, it causes network congestion and interruption at UDs. Actually, it is noticed that some UDs tend to have a common interest in downloading the same content, known as popular content. This frequently happens where UDs download a popular application, an electronic map, or streaming multimedia contents, such as videos. This consumes a huge amount of cellular system resources [6]. Therefore, D2D communication technique can be exploited to distribute the popular content as such to offload the centralized content providers (CPs) and reduce network congestion in the cellular networks. For instance, consider that a content consists of set of packets

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

p_1, p_2, and p_3 is requested by set of UDs u_1, u_2, and u_3. Suppose that the CP transmitted the requested contents to the UDs and due to channel impairments UD u_i did not receive packet p_i for $1 \leq i \leq 3$. Traditionally, these lost packets are retransmitted repeatedly from the CP to each UD until all UDs receive them correctly. As a result, the CP needs at least 3 uncoded transmissions for delivering these packets, which degrade system performance [7]. However, UDs can be either packet holders that can provide their received packets to other UDs or packet requesters that can receive the requested packets from other UDs. In our considered example, the u_1-th UD holds packets p_2 and p_3, and accordingly, it can transmit binary XOR[1] combination $p_2 \oplus p_3$ to the u_2-th and u_3-th UDs. Then, the u_2-th UD holds p_1 and can provide it to the u_1-th UD. As a result, 2 transmissions are required for delivering all packets to all UDs. Therefore, D2D techniques can be used with binary XOR combination to combine contents and transmit them to interested UDs via D2D links. As a result, the number of transmission slots is minimized and the CP's resources are offloaded. As another example, consider that a popular video content representing a frame of packets. Assume that UDs have already received some packets and requested some other packets from that frame. For continuously streaming that video without any interruption, UDs should receive their requested packets with minimum number of transmissions. For such a case, UDs can XOR the previously received packets and cooperate among them to progressively deliver requested packets to all requesting UDs with the minimum number of D2D transmissions. Such UDs cooperation and progressive packet decoding at the UDs meet the delay requirements and ensure good streaming quality.

Joint content distribution and resource allocation problem has been considered in different works, see for example, [31], [32]. In particular, the authors of [31] developed a novel framework that takes UD-centric information, such as content distribution and mobility patterns, into account to effectively deploy flying cache-enabled remote radio heads (RRHs). As such, UDs' quality-of-experience (QoE) in cloud radio access network (CRAN) is maximized. In [32], the authors solved the joint caching and resource allocation problem for a network of cache-enabled unmanned aerial vehicles (UAVs). As such, the number of stable queue UDs is maximized. However, the aforementioned works did not consider the problem of delivering a set of requested contents to a set of requesting UDs by exploiting both XOR combinations of UDs' requests and UDs cooperation over D2D links. In this work, we are interested in the content delivery problem in partially connected D2D networks where the aim is to minimize the number of D2D transmissions that are required to deliver all requested contents to all UDs. This problem is usually referred as the *completion time minimization* problem.

The completion time minimization problem was solved in different network settings for various applications. For example, Raptor codes [8] and Random Linear Network Codes (RLNC) [9] addressed the problem and achieved maximum network throughput. However, they are not suitable techniques for delivering delay-sensitive contents for real-time applications, such as online gaming, and teleconferencing. Actually, they accumulate a substantial decoding delay, meaning that these codes do not allow progressive decoding. In particular, combined contents cannot be decoded to retrieve the original data until enough number of independent transmissions are received. On the other hand, a low-complexity solution, called Instantly Decodable Network Coding (IDNC), improves throughput while allowing progressive decoding of the received contents [10]. By relying solely on binary XOR operations, IDNC ensures fast and instantaneous decodability of the transmitted contents for their intended UDs. Therefore, IDNC has been the topic of extensive research, see for example, [11]–[17]. It has been applied in several real-time broadcast applications wherein received contents need to be used at the application layer immediately to maintain a certain QoS, e.g., relay-aided networks [18], video-on-demand and multimedia streaming [19]–[25], and D2D-enabled systems [26]–[30].

The aforementioned IDNC works, for both fixed infrastructures and D2D networks, are centralized in the sense that they require a central cloud to plan XOR combinations and coordinate transmissions. For example, the authors of [29] considered the completion time minimization problem in a partially connected D2D fog radio access network (FRAN). The problem was solved by optimization with the assumption that the fog is within the transmission range of all UDs and has perfect knowledge of the network topology. Accordingly, the fog selects transmitting UDs and their XOR combinations and conveys the information to the UDs for execution. The above mentioned centralized approaches provide good performance for the decentralized system, but they bring significant challenges to the system. First, they require high computation cost at the fog units and high power consumption at each UD. Indeed, UDs need to send the status of all D2D channels to the fog after each transmission. Second, the fog requires to know the downloading history of UDs for content delivery. Recently, in [33], [34], a non-cooperative game was used to reduce the completion time for content reception. Still, the authors just considered a *fully* connected scenario that only one player transmits the contents in each transmission slot. The fully connected model is ideal in the sense that all players are connected, and each player makes its decisions individually and selfishly. Thus, it ignored the cooperative and altruistic decisions of players to further minimize the delay for delivering the contents [35].

The cooperative and altruistic decisions of players have shown the potential of coalition games in optimizing different parameters in different models [36]–[40]. For example, the tutorial in [36] classified the coalition games and demonstrated the applications of coalition games in communication networks. The authors of [37] proposed a distributed game-theoretical scheme for players cooperation in wireless

[1] The "XOR" term in this work is referred to the bit-wise Exclusive OR.

networks to maximize the sum-rate. In [40], the authors recommended that UDs can cooperatively receive the same content through multi-hop D2D communications. However, the aforementioned works are agnostic to the available side information at the network layer, i.e., requested and previously received contents by different UDs. As a result, each UD sends uncoded (i.e., without network coding) content that serves a single UD. Such side information can be exploited to efficiently select a combination of contents that can benefit a subset of interested UDs.

In addition to their investigations in uncoded existing works, coalition games for network coding-enabled networks have been extensively explored from various perspectives. Prior network coding works with coalition games aimed to maximize sum-rate, see the works in [41], [42]. In particular, the authors of [42] showed the potential of network coding in maximizing the sum-rate in network coding-aided D2D communication. Moreover, some prior works analyzed the problem of content dissemination when network coding is enabled, e.g., [43], [44]. The authors of [43] presented an efficient non-cooperative game theoretic scheme for content dissemination using RLNC. However, the above related works [43], [44] used RLNC for content dissemination. As previously mentioned, RLNC is not a suitable technique for delivering delay-sensitive contents for real-time applications of interest in this work. Recently, the authors of [45] proposed a coalition game framework for packet delivery reduction in a partially connected D2D network using IDNC. However, the authors assumed that the cluster head selection is predetermined. Moreover, all clusters are formed once at the beginning of recovery phase. This does not give flexibility to UDs to choose their own groups in the next transmission. Importantly, if no available packets in any cluster, UDs in that cluster have to stop their intra-cluster recovery phase and wait for their cluster head to perform the inter-cluster recovery phase. This results in prolonged packet delivery time. To this end, we consider a more general setting in which a *decentralized* and *distributed* framework is developed to: i) derive the rules for optimizing the selection of the transmitting player in each formed coalition, and ii) give flexibility to UDs to choose their own alliances at each transmission. As such, all packets are delivered to all UDs in only intra-coalition delivery phase with minimum completion time.

B. MOTIVATION AND CONTRIBUTIONS

Motivated by the above limitations of the existing works, our work considers partially connected D2D networks comprising several single-antenna UDs distributed in a local hotspot, and each UD is partially connected to other UDs. The completion time minimization problem is motivated by real-time applications that tolerate only small latency. In such applications, the contents, represented by a frame of packets, need to be delivered to UDs via D2D links with the minimum number of D2D transmissions. Our proposed model can be applied in different applications. For example, UDs at the edge of the service area or in dense urban areas often experience content losses from CPs due to channel impairments. Our proposed distributed scheme can be used to serve these UDs via D2D links. Moreover, in cell centers with low erasures, our proposed scheme has the potential to offload the CP's resources, e.g., time, bandwidth, and the ability to serve more UDs. The main contributions of this work are summarized as follows.

1) For partially connected D2D networks, we develop a decentralize IDNC-assisted D2D game theoretical framework to minimize the completion time while offloading the CPs. In particular, the completion time minimization problem is formulated and modeled as a coalition game. Given the difficulty of expressing the problem as a coalition game with non-transfer function (NTU), we relax it to a *coalition formation game* (CFG).
2) In the relaxed CFG, the completion time metric is written as the utility function, which is transferred to each player's[2] payoff in each coalition. We derive the rules for associating players, selecting the transmitting player, and finding its packet combination that is beneficial for a set of interested UDs in each disjoint altruistic coalition. Afterward, we develop a coalition formation distributed algorithm in each transmission slot based on merge-and-split rules.
3) The proposed coalition formation algorithm is proved to converge to a Nash-stable equilibrium, and its complexity and communication overhead are analyzed theoretically. We validate our theoretical findings using comprehensive numerical simulations, which reveal that our distributed scheme can significantly outperform existing centralized and fully distributed methods. For the presented network setups, our proposed decentralized scheme offers almost the same performance as the centralized D2D FRAN scheme.

II. SYSTEM OVERVIEW AND PROBLEM FORMULATION

A. SYSTEM OVERVIEW

Consider a D2D-enabled wireless network, shown in Fig. 1, that consists of N user devices (UDs), denoted by the set $\mathcal{N} = \{u_1, u_2, \cdots, u_N\}$. These UDs are interested in receiving a delay-sensitive content representing a frame of M packets, denoted by the set $\mathcal{P} = \{p_1, p_2, \cdots, p_M\}$. The size of the frame $\mathcal{P}$ depends on the sizes of the packet and content. This work assumes that UDs have already acquired some packets in $\mathcal{P}$ from previous transmissions, i.e., $\mathcal{H}_{u_k}$ denotes the *Has* set of packets acquired by the u_k-th UD. We assume UDs request a set of packets, which is not included in their *Has* sets, from the frame $\mathcal{P}$. This requested set will be referred as *Wants* set, i.e., $\mathcal{W}_{u_k} = \mathcal{P} \setminus \mathcal{H}_{u_k}$ is set of wanted packets for u_k-th UD. The *Has* and *Wants* sets of all UDs can be summarized in a binary $N \times M$ *state matrix* $\mathbf{S} = [s_{u_k,p_l}]$ wherein the entry $s_{u_k,p_l} = 0$ represents that the p_l-th packet

[2] In the context of game theory in this paper, each UD acts as a game player, and thus, UDs and players are used interchangeably.

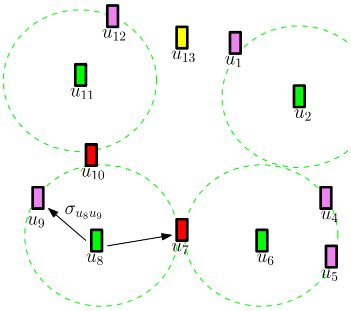

FIGURE 1. Illustration of a partially connected D2D network with 13 UDs. For simplicity, only the coverage zones of the possibly transmitting UDs u_2, u_6, u_8, and u_{11} are drawn.

is successfully received at the u_k-th UD and 1 otherwise. In order for all UDs to obtain the whole frame $\mathcal{P}$ from D2D communications, we assume that at the beginning of D2D transmissions each packet p_1, $1 \leq l \leq M$ is received by at least one UD, i.e., $\sum_{u_k \in \mathcal{U}} s_{u_k,p_l} \geq 1$ for all packets $p_l \in \mathcal{P}$. At each transmission slot, each UD can be packet requester (receiver) or packet holder (transmitter) that can provide its received packets to other requesting UDs via D2D links. When any UD receives all its requested packets, it has a large *Has* set size to generate a potential IDNC packet for many neighboring UDs. Thus, it can be selected as a potential D2D transmitter to serve a significant number of receiving UDs.

This work considers a realistic partial network topology, where UDs can only target the subset of UDs in their coverage zone, denoted by $\mathcal{Z}_{u_k}$ of the u_k-th UD. The network topology can be captured by a unit diagonal symmetric $N \times N$ adjacency matrix $\mathbf{Z}$ that represents the connectivity of the UDs such that $\mathbf{Z}_{u_k u_l} = 1$ if and only if $u_l \in \mathcal{Z}_{u_k}$. The matrix $\mathbf{Z}$ is updated after each transmission based on the connectivity index C, where C is a fractional connection ratio. We assume that no part of the network is disjoint, i.e., the matrix $\mathbf{Z}$ is connected. Otherwise, the proposed algorithm is separately applied to each independent part of the network. Upon successful reception of a packet, each UD sends an error-free acknowledgment (ACK) to all UDs in its coverage zone to update their *Has* and *Wants* sets. Therefore, maintaining a feedback matrix of neighboring UDs required low complexity and low signaling overhead than the fully connected model.

We focus only on the upper layer view of the network, where network coding is performed at the network layer, and a memory-less erasure channel abstracts the physical-layer. This abstraction is widely used in network coding literature, where a packet is either perfectly received or completely lost with certain probability [11], [13], [26], [28]–[30], [33], [34], [46]. Therefore, the physical channel between the u_k-th and the u_l-th UDs is modeled by a Bernoulli random variable whose mean $\sigma_{u_k u_l}$ indicates the packet erasure probability from u_k-th UD to u_l-th UD. We assume that these probabilities remain constant during the transmission of a single packet $p_l \in \mathcal{P}$, and they are known to all UDs. Due to the channel's asymmetry between the u_k-th and the u_l-th UDs, the equality of $\sigma_{u_k u_l}$ and $\sigma_{u_l u_k}$ is not guaranteed. To clarify the system model and its corresponding variables, Fig. 1 shows a partially connected D2D network with 13 UDs. The coverage zones of the possibly transmitting green color UDs are drawn and their targeted UDs are shown in violet color in Fig. 1. Similarly, UDs in the intersection of transmitting UDs are shown in red color, and those are located out of transmission of any transmitting UD are shown in yellow color.

We consider a slowly changing network topology, in which UDs have fixed locations during the IDNC packet transmission and change from one transmission to another. Thus, after one transmission, the UDs are moved, and all the network variables, such as the connectivity matrix and erasure probability between UDS are updated. It is important to note that in fully connected networks, each UD is connected to all other UDs in the network, and hence, it precisely knows the *Has* and *Wants* sets of all UDs. To avoid any collision in the network, only one UD is allowed to transmit an encoded packet at any transmission slot. In partial connected D2D networks, single-interface UDs can receive two transmissions from different transmitting UDs, but this causes collisions. This collision is due to two or more UDs transmit simultaneously using the same radio resource block, e.g., frequency. As such, the receiving UD cannot decode these collided packets correctly, see for examples [28]–[30], [33], [34], and their references. Therefore, UDs are grouped in disjoint and independent coalitions to avoid any collisions or conflict transmissions. In particular, UDs are randomly grouped in disjoint and independent coalitions. Then, using successive split and merge rules that explained in Section IV, UDs decide to joint/leave their coalitions based on their own preferences as well as coalitions' preferences. The whole process of grouping the UDs in disjoint coalitions is explained in details in the proposed coalition formation solution in Section IV. Considering the interference of transmissions caused by other UDs to the set of transmitting UDs can be pursued in future work.

B. INSTANTLY DECODABLE NETWORK CODING IN THE NETWORK LAYER

IDNC XOR combination enables the complete packet reception at every time slot when each scheduled UD has at most one packet in that XOR combination. Let p_{u_k} be an XOR combination sent by the u_k-th UD to the set of scheduled UDs $\mathrm{u}(\mathrm{p}_{u_k})$. Note that the packet combination $\mathrm{p}_{u_k} \subset \mathcal{H}_{u_k}$ is an element of the power set $\mathcal{X}$ of the received packets at the u_k-th UD. A transmission from the u_k-th UD is instantly decodable at the u_l-th UD if: i) it contains only one packet from the *Wants* set of the u_l-th UD and ii) the u_l-th UD is in the coverage zone of the u_k-th UD. Therefore, the set of targeted[3] UDs by u_k-th UD is expressed as $\mathrm{u}(\mathrm{p}_{u_k}) = \left\{u_l \in \mathcal{N} \,\middle|\, |\mathrm{p}_{u_k} \cap \mathcal{W}_{u_l}| = 1 \text{ and } u_l \in \mathcal{Z}_{u_k}\right\}$. In that case, the u_l-th UD can re-XOR the combination p_{u_k} with $\mathcal{H}_{u_l}$ to

[3] The term "targeted UDs" is given for a set of scheduled UDs that receives an instantly-decodable transmission.

retrieve one of its requesting packets. Hence, we say that the u_k-th UD is a transmitting UD that can provide a set of its received packets to a set of targeted UDs.

Let $\mathcal{N}_{\text{tra},t} \subset \mathcal{N}$ denote the set of transmitting UDs at the t-th transmission and $\underline{\mathtt{p}}(\mathcal{N}_{\text{tra},t}) = (\mathtt{p}_{u_1}, \cdots, \mathtt{p}_{u_{|\mathcal{N}_{\text{tra},t}|}})$ denote the packet combinations to be sent by UDs in $\mathcal{N}_{\text{tra},t}$. For simplicity, the transmission index t is often omitted when it is clear from the context. Let $\underline{\mathtt{u}}(\underline{\mathtt{p}}(\mathcal{N}_{\text{tra}})) = (\mathtt{u}(\mathtt{p}_{u_k}), \cdots, \mathtt{u}(\mathtt{p}_{u_{|\mathcal{N}_{\text{tra}}|}}))$ denote the set of targeted UDs by the transmitting UDs wherein $u_l \in \mathtt{u}(\mathtt{p}_{u_k}(\mathcal{N}_{\text{tra}}))$ implies that $|\mathcal{W}_{u_l} \cap \mathtt{p}_{u_k}(\mathcal{N}_{\text{tra}})| = 1$ and $\{u_l\} \cap \mathcal{Z}_{u_k} \cap \mathcal{Z}_{u_m} = \delta_{u_k u_m}\{u_l\}$ for all transmitting UDs $u_m \in \mathcal{N}_{\text{tra}}$ wherein $\delta_{u_k u_m}$ is the Kronecker symbol. Mathematically, $\delta_{u_k u_m}\{u_l\} = u_l$ if $u_k = u_m$ and 0 otherwise.

Definition 1: The completion time of the u_k-th UD, denoted by $\mathcal{T}_{u_k}$, is the number of D2D transmissions required to get all its requested packets in $\mathcal{W}_{u_k}$. The overall completion time $\mathcal{T} = \max_{u_k \in \mathcal{N}}\{\mathcal{T}_{u_k}\}$ represents the time required for all UDs to get all the packets.

We use IDNC to minimize the completion time required to complete the delivery of all packets to all UDs in the partially connected D2D network. Given that the direct minimization of the completion time is intractable [30], we follow [15] in reducing the completion time by controlling the decoding delay.

Definition 2: The decoding delay of the u_l-th UD, denoted by $\mathcal{D}_{u_l}$, increases by one unit if and only if $\mathcal{W}_{u_l} \neq \varnothing$ and the u_l-th UD receives a non-instantly decodable transmission from any transmitting UD in $\mathcal{N}_{tra}$. The decoding delay $\mathcal{D}$ is the sum of all individual delays.

C. COMPLETION TIME MINIMIZATION PROBLEM FORMULATION

In this subsection, we formulate the distributed completion time reduction problem in IDNC-enabled D2D network. Let $\underline{N}_w$ be a binary vector of size N whose u_k-th index is 1 if the u_k-th UD has non-empty *Wants* set, i.e., $\mathcal{W}_{u_k} \neq \varnothing$ and 0 otherwise, and let $\underline{\bar{\mathtt{u}}}(\underline{\mathtt{p}}(\mathcal{N}_{\text{tra}})) = \underline{1} - \underline{\mathtt{u}}(\underline{\mathtt{p}}(\mathcal{N}_{\text{tra}}))$ be the set of the non-targeted UDs by the encoded packets $\underline{\mathtt{p}}(\mathcal{N}_{\text{tra}})$. The different erasure occurrences at the t-th transmission slot are denoted by $\boldsymbol{\omega} : \mathbb{Z}_+ \rightarrow \{0, 1\}^{N \times N}$ with $\boldsymbol{\omega}_t = [Y_{u_k u_l}]$, for all $(u_k, u_l) \in \mathcal{N}^2$, where $Y_{u_k u_l}$ is a Bernoulli random variable equal to 0 with probability $\sigma_{u_k u_l}$.

Let $\underline{n}_t = (n^{[u_1]}, n^{[u_2]}, \cdots, n^{[u_N]})$ be a binary vector of length N whose $n^{[u_k]}$-th element is equal to 1 if u_k-th UD is transmitting, i.e., $\|\underline{n}\|_1 = |\mathcal{N}_{\text{tra}}|$. Likewise, let $\underline{\mathcal{D}}(\underline{n}_t)$ be the decoding delay experienced by all UDs in the t-th transmission slot. In particular, $\underline{\mathcal{D}}(\underline{n}_t)$ is a metric that quantifies the ability of the transmitting UDs to generate innovative packets for all the targeted UDs. This metric increases by one unit for each UD that still requests packets and successfully receives a non-useful transmission from any transmitting UD in $\mathcal{N}_{\text{tra}}$ or for a transmitting UD that still requests some packets. Let $\underline{\mathcal{I}} = (\mathcal{I}^{[u_1]}, \mathcal{I}^{[u_2]}, \cdots, \mathcal{I}^{[u_N]})$ be a binary vector of size N whose $\mathcal{I}^{[u_l]}$ entry is 1 if the u_l-th UD is hearing more than one transmission from the set $\mathcal{N}_{\text{tra}}$, i.e., $u_l \in \mathcal{Z}_{u_k} \cap \mathcal{Z}_{u_m}$ where $u_k \neq u_m \in \mathcal{N}_{\text{tra}}$ and 0 otherwise, and let $\underline{\mathcal{O}} = (\mathcal{O}^{[u_1]}, \mathcal{O}^{[u_2]}, \cdots, \mathcal{O}^{[u_N]})$ be a binary vector of size N whose $\mathcal{O}^{[u_l]}$ element is 1 if the u_l-th UD is out of transmission range of any transmitting UD in $\mathcal{N}_{\text{tra}}$, i.e., $u_l \notin \mathcal{Z}_{u_k}, \forall u_k \in \mathcal{N}_{\text{tra}}$ and 0 otherwise. The used notations and variables in this paper are summarized in Table 1.

Given the aforementioned configurations, the overall decoding delays $\underline{\mathbb{D}}(\underline{n}_t)$ experienced by all UDs, since the beginning of the delivery phase until the t-th transmission, can be expressed as follows.

$$\underline{\mathbb{D}}(\underline{n}_t) = \underline{\mathbb{D}}(\underline{n}_{t-1}) + \begin{cases} \underline{N} & \text{if } \|\underline{n}_t\|_1 = 0 \\ \underline{\mathcal{I}} + \underline{\mathcal{O}} + \underline{n}_t + \underline{\mathcal{D}}(\underline{n}_t) & \text{otherwise.} \end{cases} \tag{1}$$

As mentioned, the completion time is a difficult and intractable metric to optimize. However, the authors in [15] proposed to approximate such metric by the *anticipated* completion time which can be computed at each transmission using the decoding delay. The anticipated completion time that uses the decoding delay in (1) is defined as follows.

Definition 3: The anticipated completion time of the u_k-th UD is defined by the following expression

$$\mathcal{T}_{u_k}(\underline{n}_t) = \frac{|\mathcal{W}_{u_k,0}| + \mathbb{D}_{u_k}(\underline{n}_t) - \mathbb{E}[\sigma_{u_k}]}{1 - \mathbb{E}[\sigma_{u_k}]}, \tag{2}$$

where $|\mathcal{W}_{u_k,0}|$ is the number of the requested packets in the Wants set of u_k-th UD at the beginning of delivery phase and $\mathbb{E}[\sigma_{u_k}]$ is the expected erasure probability linking u_k-th UD to the other UDs in its coverage.

Clearly, (2) represents the number of D2D transmissions that are required for the u_k-th UD to receive all the packets in $\mathcal{W}_{u_k}$. It depends on the initial set of requested packets by each UD, decoding delay, and packets' erasures. Essentially, the decoding delay is a metric that quantifies the ability of the D2D transmitters to generate innovative packets for receiving UDs. As previously stated, this metric increases by one unit for each UD that still requested packets and successfully receives a non-useful transmission.

The overall anticipated completion time can be written as $\mathcal{T}(\underline{n}_t) = \max_{u_k}(\mathcal{T}_{u_k}(\underline{n}_t)) = \|\underline{\mathcal{T}}(\underline{n}_t)\|_\infty$. Therefore, the anticipated completion time minimization problem at the t-th transmission slot in IDNC-enabled partially D2D network can be written as follows.

$$\min_{\substack{\underline{n}_t \in \{0,1\}^N \\ \underline{\mathtt{p}}(\mathcal{N}_{\text{tra}}) \in \{0,1\}^M}} \|\underline{\mathcal{T}}(\underline{n}_t)\|_\infty. \tag{3}$$

Unlike a fully connected model that only requires optimization over a single transmitting UD and its corresponding packet combination, a partially connected model needs to select the set of transmitting UDs $\mathcal{N}_{\text{tra}}$ and the encoded packets $\underline{\mathtt{p}}(\mathcal{N}_{\text{tra}})$. As such, the probability of increasing the anticipated completion time is minimized.

TABLE 1. Variables and parameters of the system.

Variable	Definition
$\mathcal{N}$	Set of N UDs
$\mathcal{P}$	Set of M packets
$\mathcal{W}_{u_k}$	Set of wanted packets by the u_k-th UD
$\mathcal{H}_{u_k}$	Set of acquired packets by the u_k-th UD
$\mathcal{Z}_{u_k}$	Set of nearby UDs in the coverage zone of the u_k-th UD
$\sigma_{u_k u_l}$	Packet erasure probability from the u_k-th UD to the u_l-th UD
p_{u_k}	The encoded packet of the u_k-th transmitting UD
$\mathrm{u}(\mathrm{p}_{u_k})$	Set of targeted UDs by the u_k-th UD
$\mathcal{N}_{\text{tra}}$	Set of transmitting UDs
$\underline{\mathrm{p}}(\mathcal{N}_{\text{tra}})$	Packet combinations to be sent by UDs in the set $\mathcal{N}_{\text{tra}}$
$\underline{\mathrm{u}}(\underline{\mathrm{p}}(\mathcal{N}_{\text{tra}}))$	Set of all targeted UDs by the transmitting UDs
$\underline{n}_t$	Binary vector of the transmitting UDs
$\underline{\mathcal{D}}(\underline{n}_t)$	Decoding delay experienced by all UDs
$\underline{\mathcal{I}}$	Binary vector for UDs hearing more than one transmission from the set $\mathcal{N}_{\text{tra}}$
$\underline{\mathcal{O}}$	Binary vector for UDs out of transmission range of any transmitting UD in $\mathcal{N}_{\text{tra}}$

D. EXAMPLE OF IDNC TRANSMISSIONS IN A PARTIALLY CONNECTED D2D NETWORK

This section illustrates the aforementioned definitions and concepts through a simple example. Consider a simple partially connected D2D network containing 6 UDs and a frame $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$ as illustrated in Fig. 2. The acquired and wanted packets of all UDs are given on the left part of Fig. 2, and the coverage zone of each UD is represented by edges. For ease of analysis, we assume error-free transmissions.

Assume that the u_1-th and the u_6-th UDs transmit $\mathrm{p}_{u_1} = p_3 \oplus p_4$ and $\mathrm{p}_{u_6} = p_1 \oplus p_4$ in the first transmission slot to the sets $\mathrm{u}(\mathrm{p}_{u_1}) = \{u_2, u_3, u_5\}$ and $\mathrm{u}(\mathrm{p}_{u_6}) = \{u_4, u_5\}$, respectively. In the second transmission slot, assume that the u_4-th and the u_1-th UDs transmit $\mathrm{p}_{u_4} = p_2$ and $\mathrm{p}_{u_1} = p_2 \oplus p_4$ to the sets $\mathrm{u}(\mathrm{p}_{u_4}) = \{u_6\}$ and $\mathrm{u}(\mathrm{p}_{u_1}) = \{u_2, u_5\}$, respectively. The decoding delay experienced by the different UDs is given as follows.

- The u_5-th UD experiences one unit delay as it is in the intersection of the coverage zone of transmitting UDs u_1 and u_6. In other words, u_5 is in collision, i.e., $u_5 \in \underline{\mathcal{I}}$. Thus, the u_5-th UD cannot decode packet p_6 transmitted by the u_6-th UD.
- The u_6-th UD experiences one unit of delay as it transmits in the first transmission slot.

Under this scenario, we have the following assumption.

- **First transmission slot:** $\underline{N}_w = (0\ 1\ 1\ 1\ 1\ 1)$, the set of transmitting UDs $\mathcal{N}_{\text{tra},1} = \{u_1, u_6\} = \underline{n}_1 = (1\ 0\ 0\ 0\ 0\ 1)$, the corresponding encoded packets $\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},1}) = (\mathrm{p}_{u_1}, \mathrm{p}_{u_6})$, and the set of targeted UDs $\underline{\mathrm{u}}(\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},1})) = \{(u_2, u_3), (u_4)\}$. The set of UDs hearing more than one transmission $\underline{\mathcal{I}} = (0\,0\,0\,0\,1\,0)$, and the set of UDs that out of transmission range of any UD in $\mathcal{N}_{\text{tra},1}$ is $\underline{\mathcal{O}} = \underline{0}$. The decoding delay experienced by all UDs is $\underline{\mathcal{D}}(\underline{n}_1) = (0\,0\,0\,0\,1\,1)$. The accumulative decoding delay is $\underline{\mathbb{D}}(\underline{n}_1) = (0\,0\,0\,0\,1\,1)$.

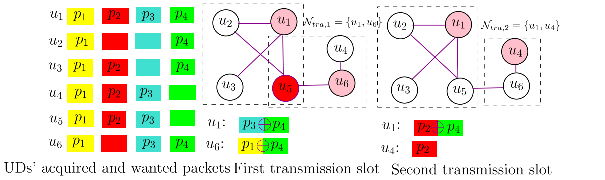

FIGURE 2. A partially connected D2D network containing 6 UDs and 4 packets.

- **Second transmission slot:** $\underline{N}_w = (0\ 1\ 0\ 0\ 1\ 1)$, the set of transmitting UDs $\mathcal{N}_{\text{tra},2} = \{u_1, u_4\} = \underline{n}_2 = (1\ 0\ 0\ 1\ 0\ 0)$, the corresponding encoded packets $\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},2}) = (\mathrm{p}_{u_1}, \mathrm{p}_{u_4})$, and the set of targeted UDs $\underline{\mathrm{u}}(\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},2})) = \{(u_2, u_5), (u_6)\}$. The set of UDs hearing more than one transmission $\underline{\mathcal{I}} = \underline{0}$, and the set of UDs that out of transmission range of any UD in $\mathcal{N}_{\text{tra},2}$ is $\underline{\mathcal{O}} = \underline{0}$. The decoding delay is $\underline{\mathcal{D}}(\underline{n}_2) = \underline{0}$ and the accumulative decoding delay $\underline{\mathbb{D}}(\underline{n}_2) = (0\,0\,0\,0\,1\,1)$.
- The completion time of all UDs after the second transmission is $\mathcal{T} = (0\ 2\ 1\ 1\ 2\ 2)$. Thus, the maximum completion time is 2 transmission slots which represents the overall completion time for all UDs to get their requested packets, i.e., $\underline{N}_w = \underline{0}$.

III. DISTRIBUTED COMPLETION TIME MINIMIZATION AS A COALITION GAME

Our main goal is to develop a distributed framework that models the collaborations among the UDs of IDNC-enabled partially connected D2D network. In order to do that, we use game theory because it involves a set of players that interact with each other to form a coalition without any coordination from the CP. The key idea of forming coalitions is to study the cooperative behavior of UDs in coalitions. In particular, UDs are formed coalitions to minimize their completion times compared to when they do not form coalitions as will be discussed in Section IV. With coalitions and IDNC content delivery among players, the overall completion time can be reduced.

In this section, we first model the completion time problem in IDNC-enabled partially connected D2D networks using a coalition game [35]. Afterward, we define and provide fundamental concepts in coalition games. These concepts are used in Section IV to derive the proposed distributed completion time reduction solution.

A. COMPLETION TIME MINIMIZATION AS A COALITION GAME

We use a coalition game that suites our model to formulate the completion mentioned above time problem. In particular, our problem is modeled as a coalition game with a non-transferable utility (NTU) [35].

Definition 4: A coalition game with a non-transferable utility is defined as a pair $(\mathcal{N}, \phi)$, where $\mathcal{N}$ is the set of players consisting of N UDs and ϕ is a real function such that for every coalition $\mathcal{S}_s \subseteq \mathcal{N}$, $\phi(\mathcal{S}_s)$ is the payoff that coalition $\mathcal{S}_s$ receives which cannot be arbitrarily apportioned between its players.

For the considered D2D completion time problem with coalition games, packets are transmitted simultaneously from a set of cooperative transmitting UDs $\mathcal{N}_{\text{tra}}$. Each transmitting UD serves a group of interested and nearby UDs that constitutes a coalition. In the coalition game, each UD acts as a game player and aims to join a coalition such that: i) it can receive packets and minimize its completion time, and ii) it does not increase the completion time of its alliances in the coalition. Therefore, for each coalition, we need to determine the transmitting player and its packet combination that is beneficial to a set of interested players. As such, we minimize the increasing of the completion time.

Let $\phi(\mathcal{S}_s) = (\phi_{u_1}(\mathcal{S}_s), \cdots, \phi_{u_{|\mathcal{S}_s|}}(\mathcal{S}_s))$ define a tuple wherein element $\phi_{u_k}(\mathcal{S}_s)$ represents the payoff of u_k-th UD in coalition $\mathcal{S}_s$, where $\mathcal{S}_s \subseteq \mathcal{N}$, and $|\mathcal{S}_s|$ is the total number of players in $\mathcal{S}_s$. We use s as the subscript to identify coalitions. The $|\mathcal{S}|$-dimensional vector represents the family of real vector payoffs of coalition $\mathcal{S}_s$, which is denoted by $\underline{\phi}(\mathcal{S}_s)$. Consequently, by adopting the cooperative D2D completion model described in the previous section, the total payoff of any coalition $\mathcal{S}_s \subseteq \mathcal{N}, \forall s = \{1, \cdots, m\}$ is given by

$$\phi(\mathcal{S}_s) = \max_{u_k}(\phi_{u_k}(\mathcal{S}_s)) = \|\underline{\phi}(\mathcal{S}_s)\|_\infty, \tag{4}$$

where $\phi_{u_k}(\mathcal{S}_s)$ is the payoff of u_k-th UD in a coalition $\mathcal{S}_s$ which is in our problem given by

$$\phi_{u_k}(\mathcal{S}_s) = -\|\mathcal{T}_{u_k}(\underline{n}_t)\|_\infty - \|\mathbb{D}_{u_k}(\underline{n}_t) - \mathbb{D}_{u_k}(\underline{n}_{t-1})\|_1. \tag{5}$$

The payoff function in (4) represents the total payoff that a coalition receives due to self-organize players. For a player $u_k \in \mathcal{S}_s$, the first term in (5) represents the maximum anticipated completion time among players in $\mathcal{S}_s$ that is defined in (2). Similarly, the second term in (5) represents the augmentation of the sum decoding delay that is defined in (1). Therefore, players in coalitions prefer to increase the payoff in (5) by minimizing the anticipated completion time through controlling the decoding delay. Note that the negative sign indicates that the payoff of a coalition is inversely proportional to the completion time.

Property 1: The proposed D2D completion time cooperative problem is modeled as a coalition game with NTU $(\mathcal{N}, \phi)$ where $\mathcal{N}$ is the set of players and ϕ is the payoff function given by (4).

Proof: From the nature of definition 1 and definition 2, the u_k-th player has its own unique anticipated completion time and decoding delay, and, thus, it has a unique payoff $\phi_{u_k}(\mathcal{S}_s)$ within a coalition $\mathcal{S}_s$. Therefore, the payoff function in (4) cannot be arbitrarily apportioned between coalition's players. Thus (4) is considered as an NTU. Further, the overall completion time is the maximum individual completion times of the players regardless of the coalition. In other words, the dependency of $\phi(\mathcal{S}_s)$ in any coalition structure is not only on packet delivery of players inside $\mathcal{S}_s$, but also on packet delivery outside $\mathcal{S}_s$, which concludes that the proposed game model is NTU game. ■

Although cooperation generally reduces the payoffs of players [35], it is limited by inherent information exchange cost that needs to be paid by the players when acting cooperatively. Consequently, for any coalition $\mathcal{S}_s \subseteq \mathcal{N}$, players need to exchange information for cooperation, which is an increasing function of the coalition size. The problem becomes severe when all players are in the same coalition, i.e., grand coalition (GC). However, given the realistic scenario of a partially connected network where each UD has limited coverage zone, it is highly likely that when attempting to form the GC, one of these scenarios will hold:

- There exists a pair of the u_k-th, the u_l-th players in $\mathcal{N}$ that are distant enough to receive packets within the GC. Thus, they have no incentive to join the grand coalition.
- There exists a player $u_k \in \mathcal{N}$ with a payoff in GC $\phi_{u_k}(\mathcal{N})$ that is less than its payoff in any coalition $\phi_{u_k}(\mathcal{S}_s)$. Hence, this player has an incentive to deviate from the GC.

Further, the limited coverage players in the considered partially connected D2D model would most likely form small coalitions' sizes, not large coalitions' sizes. Therefore, the GC of all players is *seldom* formed, and the cooperation cost due to forming small coalitions' sizes would not have a significant impact on the payoff functions.

The aforementioned discussion emphasizes that the proposed $(\mathcal{N}, \phi)$ game is classified as a coalition formation game (CFG) [36], where limited coverage-zone players form several independent disjoint coalitions. Hence, classical solution concepts for coalition games, such as the core [35], may not apply to our problem. In summery, the proposed coalition game $(\mathcal{N}, \phi)$ is a CFG, where the objective is to develop an algorithm for forming coalitions.

B. COALITION FORMATION CONCEPTS

This section provides some concepts of coalition formation games that will be used in the next subsection. CFG, a subclass of coalition games, has been a topic of high interest in game theory research [36], [37], [39]. The fundamental approach in coalition formation games is to allow players in the formation set to join or leave a coalition based on a well-defined and most suitable *preference* for NTU games, i.e., *Pareto Order*. Pareto Order is the basis of many existing coalition formation concepts, e.g., the merge-and-split algorithm [38].

Definition 5: A coalition structure, denoted as Ψ, *is defined as* $\Psi = \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ *for* $1 < |\mathcal{S}_m| < |\mathcal{N}|$ *independent disjoint coalitions* $\mathcal{S}_m$ *of* Ψ.

One can see from definition 5 that different coalition structures may lead to different system payoffs as each coalition structure Ψ has its unique payoff $\phi(\Psi)$. These differences in Ψ and their corresponding payoffs $\phi(\Psi)$ are usually ordered through a comparison relationship. In the coalition game literature, see for example, [38], comparison relationships based on orders are divided into individual value orders and coalition value orders. Individual value order implies that comparison is performed based on the players' payoffs. This is referred to as the Pareto Order. In particular, in such order, no player is willing to move to another coalition when it has a negative effect on the payoff of that coalition. In other words, the payoff of players can be worse off after the new player joins. This is known as selfish behavior. Coalition order implies that two coalition structures are compared based on the payoff of the coalitions in these coalition structures. This is known as a utilitarian order and is denoted by $\triangleright$. In other words, the notation $\Psi_2 \triangleright \Psi_1$ means that $\phi(\Psi_2) > \phi(\Psi_1)$. Subsequently, the definition of the preference operator that considered in this paper is given as follows.

Definition 6: A preference operator $\triangleright$ *is defined for comparing two coalition structures* $\Psi_1 = \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ *and* $\Psi_2 = \{\mathcal{R}_1, \cdots, \mathcal{R}_n\}$ *that are partitions of the same set of players* $\mathcal{N}$. *The notation* $\Psi_2 \triangleright \Psi_1$ *denotes that players in* $\mathcal{N}$ *are preferred to be in* Ψ_2 *than* Ψ_1.

IV. PROPOSED FULLY DISTRIBUTED SOLUTION

A. COALITION FORMATION CONSTRAINTS

Let $\mathcal{N}_s$ be the set of all associated players in the $\mathcal{S}_s$-th coalition and $\mathcal{N}_{s,w}$ the subset of $\mathcal{N}_s$ that have non-empty *Wants* set. Let $\mathcal{M}_s$ be the subset of packets that in the *Has* set of each player in $\mathcal{N}_s$, which defined as $\mathcal{M}_s = \bigcup_{u_k \in \mathcal{N}_s} \mathcal{H}_{u_k}$. Let $\mathtt{S}_s$ denote the set of all neighboring coalitions to the $\mathcal{S}_s$-th coalition. For the $\mathcal{S}_s$-th coalition, the $u_{n_s^*}$-th transmitting UD is the one that can achieve the least expected increase in the completion time. According to the analysis available in [33], [34], a transmitting UD $u_{n_s^*}$ and its packet combination $\mathtt{p}_{u_{n_s^*}}$ sent to the set of UDs $\mathtt{u}(\mathtt{p}_{u_{n_s^*}})$ can be obtained by solving the following problem

$$u_{n_s^*} = \arg\max_{u_n \in \mathcal{N}_{\text{tra},s} \setminus \mathcal{L}_s} |\mathcal{Z}_{u_n} \cap \mathcal{N}_{s,w}| \\ + \max_{\mathtt{p}_{u_n} \in \underline{\mathtt{p}}(\mathcal{N}_{\text{tra},s})} \sum_{u_l \in \mathcal{L}_s \cap \mathtt{u}(\mathtt{p}_{u_n})} \log \frac{1}{\sigma_{u_n u_l}}, \quad (6)$$

where $\mathcal{N}_{\text{tra},s}$ is the set of players in coalition $\mathcal{S}_s$ that are not in any coverage zone of all other players in $\mathtt{S}_s$ and $\mathcal{L}_s$ is the set of critical players that can potentially increase the overall payoff of the coalition $\mathcal{S}_s$. This set characterizes the players based on their anticipated completion times to give them priority to be targeted in the next transmission. In other words, $\mathcal{L}_{s,t}$ contains players that will potentially increase the maximum anticipated completion time if they are not targeted in the t-th transmission. It can be expressed as

$$\mathcal{L}_{s,t} = \Big\{ u_k \in \mathcal{N} \cap \mathcal{N}_{s,w} \Big| \mathcal{T}_{u_k}(\underline{n}_{t-1}) + \frac{1}{1 - \mathbb{E}[\sigma_{u_k}]} \\ \geqslant \|\underline{\mathcal{T}}(\underline{n}_{t-1})\|_\infty \Big\}. \quad (7)$$

The set of targeted players in coalition $\mathcal{S}_s$ when device $u_{n_s^*}$ transmits the combination $\mathtt{p}_{u_{n_s^*}}$ is

$$\mathtt{u}(\mathtt{p}_{u_{n_s^*}}) = \Big\{ u_k \in \mathcal{S}_s \, \big| |\mathtt{p}_{u_{n_s^*}} \cap \mathcal{W}_{u_k}| = 1 \text{ and } \mathbf{Z}_{u_{n_s^*} u_k} = 1 \Big\}.$$

With the aforementioned variable definitions, we can reformulate the completion time minimization problem in IDNC-based partially connected D2D network per coalition at each transmission slot as follows

$$\min_{\substack{\underline{n}_t \in \{0,1\}^{|\mathcal{N}_s|} \\ \underline{\mathtt{p}} \in \{0,1\}^{|\mathcal{M}_s|}}} \phi(\mathcal{S}_s) \quad (8a)$$

$$s.\,t.\ |\mathtt{u}(\mathtt{p}_{u_{n_s^*}})| \geqslant 1, \quad (8b)$$

$$\mathtt{u}(\mathtt{p}_{u_{n_s^*}}) \cap \mathtt{u}(\mathtt{p}_{n_v^*}) = \varnothing, \forall u_{n_s^*} \neq u_{n_v^*} \in \mathtt{S}_s. \quad (8c)$$

Constraint (8b) says that the number of targeted players in each coalition must be more than one to ensure that at each transmission at least one player is benefiting. Constraint (8c) states that all the targeted players should not experience any collision. To find the optimal solution to the problem in (8), we need to search over all the sets of optimal player-coalition associations, their different erasure probabilities, players' actions, and their optimal IDNC packets. As pointed out in [29] for a centralized fog system, this is an intractable problem. Further, the solution to (8) must go through the players' decisions to join/leave a coalition at each stage of the game. To seek an efficient solution to (8) that is capable of achieving significant completion time reduction, a distributed coalition formation algorithm is developed.

B. A DISTRIBUTED COALITION FORMATION ALGORITHM

We develop a distributed coalition formation algorithm to obtain the minimum completion time of UDs. The key mechanism is to allow players in the coalition formation process to make individual decisions for selecting potential coalitions at any transmission slot. We first define two rules of merge-and-split that allow the modification of Ψ of the set $\mathcal{N}$ players as follows.

Definition 7 (Merge Operation): Any set of coalitions $\{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ *in* Ψ_1 *can be merged if and only if* $(\bigcup_{s=1}^{m} \mathcal{S}_s, \Psi_2) \triangleright (\{\mathcal{S}_1, \cdots, \mathcal{S}_m\}, \Psi_1)$. *Thus,*

$\{\mathcal{S}_1, \cdots, \mathcal{S}_m\} \rightarrow \{\bigcup_{s=1}^{m} \mathcal{S}_s\}$ and $\Psi_1 \rightarrow \Psi_2$, where $\bigcup_{s=1}^{m} \mathcal{S}_s$ and Ψ_2 are the new set of coalitions and the new coalition structure after the merge operation, respectively.

Definition 8 (Split Operation): Any set of coalitions $\bigcup_{s=1}^{m} \mathcal{S}_s$ in Ψ_1 can be split if and only if $(\{\mathcal{S}_1, \cdots, \mathcal{S}_m\}, \Psi_2) \triangleright (\bigcup_{s=1}^{m} \mathcal{S}_s, \Psi_1)$. Thus, $\{\bigcup_{s=1}^{m} \mathcal{S}_s\} \rightarrow \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ and $\Psi_1 \rightarrow \Psi_2$, where $\{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ and Ψ_2 are the new set of coalitions and the new coalition structure after the split operation, respectively.

The merge rule implies that two coalitions merge if their merger can benefit not only the players in the united alliance but also benefit the overall coalition structure value, i.e., the whole completion time. On the other hand, a coalition split into smaller ones if its splitter coalitions enhance at least the payoff of one player in that coalition. Therefore, using these two well-known rules, we present a distributed algorithm to solve the completion time minimization problem in (3) as follows.

Step 1 (Coalition Members Discovery): Players in the initialized random coalition structure Ψ_{ini} discover their neighbors by utilizing one of different well-known neighbor discovery schemes, e.g., those used in wireless networks [48]. For example, the u_k-th player broadcasts a message consisting of two segments; each segment consists of one byte. While the first byte indicates the number of players in its coverage zone $\mathcal{Z}_{u_k}$, the second byte indicates the u_k's completion time. Further, players use the collected aforementioned information to decide on the one who: i) has a large *Has* set that can serve a large number of players in its coverage zone, and ii) not in the coverage zone of any player in any other coalitions in Ψ_{ini}. However, if such a player does not exist, the size of the coalition is increased until that player exists. To summarize, the $u_{n_s^*}$-th selected transmitting player in $\mathcal{S}_s$-th coalition should satisfy (8b) and (8c) and can be obtained by solving problem (6). Afterward, each player evaluates its potential payoff as in (5) to make an accurate decision in step 2. The $u_{n_s^*}$ selected player in $\mathcal{S}_s$-th coalition ($\forall \mathcal{S}_s \in \Psi_{\text{ini}}$) will do the analysis in step 2. This step significantly reduces the search space for associating the players to the coalitions.

Step 2 (Coalition Formation for Players' Association): This step optimizes the selection of the transmitting players and their packet combinations in step 1 through many successive split-and-merge rules between coalitions. Therefore, step 2 is to associate players to potential neighboring coalitions, select the transmitting player, and find its packet combination, which can be accomplished by the following. In this step, the index is updated to $\tau = \tau + 1$. The merge rules are implemented by checking the merging possibilities of each pair of neighboring coalitions $\mathcal{S}_s$ and $\mathcal{S}_v$. Initially, $\Psi_\tau = \Psi_{\text{ini}}$. Thus, a coalition $\mathcal{S}_s \in \Psi_\tau$ can decide to merge with another coalition $\mathcal{S}_v \in \mathbb{S}_s$ to form a new coalition $\mathcal{S}_j$. As such, the resulting structure guarantees the two merge conditions (MC) defined as follows.

- MC1: There exists at least one player satisfies (8b) and (8c).
- MC2: At least one player in the merged coalition can enhance its individual payoff without affecting the payoffs of all the remaining players.

After all the coalitions have made their merge decisions based on the players' preferences, the merge rules end. This results in the updated coalition structure Ψ_τ. Similarly, the split rules are performed on the players that do not benefit from being a member of a certain coalition. In other terms, coalition $\mathcal{S}_s \in \Psi_\tau$ can be split into coalitions of smaller sizes as long as the splitter coalitions guarantee the following split conditions (SC).

- SC1: At least one player can strictly enhance its payoff without affecting the payoffs of all the remaining players.
- SC2: In each split coalition, there exists at least one player satisfying (8b) and (8c).

At the end of the split rules, the coalition structure Ψ_1 is updated. The index τ is updated along with a sequence of merge-and-split rules which take place in a distributed manner. Such sequence continues based on the resulting payoff of each player and coalition. It ends when there are no further merge-and-split rules required in the current coalition structure Ψ_τ, which is the final converged coalition structure Ψ_{fin}.

Step 3 (IDNC Packet Transmissions:) Each transmitting UD in each coalition broadcasts its IDNC packet to all UDs in its coverage zone. Accordingly, each targeted UD by any transmitter sends ACK to all its neighboring UDs indicating its packet reception status. Thus, the *Has* and *Wants* sets of all UDs in the system are updated.

The coalition formation algorithm is distributively executed at each transmission slot and summarized in Algorithm 1. It is repeated until all packets are delivered to all UDs, as presented in Algorithm 2.

Some remarks on executing Algorithm 1 are given below.

- The merge-and-split rules enumerate only neighboring coalitions, and this does not necessarily need significant computations. To further reduce the computations, the players of a coalition $\mathcal{S}_s$ can avoid merging with other neighboring coalition $\mathcal{S}_v$ if the payoffs of the players in both coalitions are equal $\phi_{u_k}(\mathcal{S}_s) = \phi_{u_l}(\mathcal{S}_v)$, $\forall u_k \in \mathcal{S}_s$ and $\forall u_l \in \mathcal{S}_v$.
- Forming coalitions only one time, i.e., at the beginning of the transmission round, is not guaranteed to deliver all packets to all players. This is because players in each formed coalition have only some packets and do not have the requested packets of other players in other coalitions. For delivering all the requested packets to all players (i.e. UDs), each coalition is formed, at each transmission slot, based on the individual preference of its alliances and irrespective of the *Has* sets of its alliances. Thus, each transmitting player has delivered

Algorithm 1 Coalition Formation Algorithm

```
Input: $\mathcal{N}, \mathcal{P}, \mathcal{H}_{u_k}, \mathcal{W}_{u_k}, \mathcal{Z}_{u_k}, \forall u_k \in \mathcal{N}$.;
Initialization: $\Psi_{\text{ini}} = \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}, \tau = 0$,
$\Psi_\tau = \Psi_{\text{ini}}$.;
Step 1: Coalition Members Discovery;
  • Each player discovers its neighboring players.;
    for each $\mathcal{S}_s \in \Psi_{ini}, \forall s = \{1, 2, \cdots, m\}$ do
      Select the transmitting players $\mathcal{N}_{\text{tra},s}$ that satisfying
      (8b) and (8c).;
      Solve (6) to find $u_{n_s^*}$ and its packet combination
      $\mathrm{p}_{u_{n_s^*}}$.;
      Calculate $\phi_{u_k}(\mathcal{S}_s)$ as in (5), $\forall u_k \in \mathcal{N}_s$.;
    end for
Step 2: Coalition Formation;
  • The optimization target in coalition $\mathcal{S}_s$ is
      $\min_{\underline{n}_t \in \{0,1\}^{|\mathcal{N}_s|},\ \underline{\mathrm{p}} \in \{0,1\}^{|\mathcal{M}_s|}} \phi(\mathcal{S}_s)$.;
  • Obtain player's assignments based on the two main
    rules of merge and split:;
    repeat
      Update $\tau = \tau + 1$.;
      for each $\mathcal{S}_s \in \Psi_{\tau-1}, \forall s = \{1, 2, \cdots, m\}$ do
        $u_{n_s^*}$ analyzes all possible merge rules with $\mathrm{S}_s$.;
        If a merge occurs:
            1) Update $\Psi_{\tau-1}$.;
            2) Solve (6) to update $\mathcal{N}_{\text{tra},s}$ and $u_{n_s^*}$.;
        Set $\Psi_\tau = \Psi_{\tau-1}$.
      end for
      for each $\mathcal{S}_s \in \Psi_\tau, \forall s = \{1, 2, \cdots, m\}$ do
        $u_{n_s^*}$ analyzes all possible split rules.;
        If a split occurs:
            1) Update $\Psi_\tau$.;
            2) Update $\mathcal{N}_{\text{tra},s}$ and $u_{n_s^*}$ by solving (6).;
      end for
    until No further merge nor split rules;
Output The convergence coalition structure $\Psi_{\text{fin}} = \Psi_\tau$.;
Step 3: IDNC Packet Transmission;
  • Each $u_{n_s^*}$ in each coalition $\mathcal{S}_s$ broadcasts $\mathrm{p}_{u_{n_s^*}}$ to all
    players in its coverage zone, $\forall s = \{1, 2, \cdots, m\}$.
```

Algorithm 2 Overall Completion Time Approach for Solving Problem (3)

```
Data: $\mathcal{N}, \mathcal{P}, \mathcal{H}_{u_k}, \mathcal{W}_{u_k}, \mathcal{Z}_{u_k}, \mathcal{T}_{u_k} = 0, \mathcal{D}_{u_k} = 0$,
$\forall u_k \in \mathcal{N}$ and $\epsilon$.;
Initialize: Transmission slot $t = 1$
Repeat:
  • Execute Algorithm 1 to obtain $\mathcal{N}_{\text{tra},t}$, $\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},t})$, and
    $\underline{\mathrm{u}}(\underline{\mathrm{p}}(\mathcal{N}_{\text{tra},t}))$ in $\Psi_{\text{fin}}$.;
  • For $\forall u_{n_s^*} \in \mathcal{N}_{\text{tra},t}$ do;
      – Each $u_k \in \mathrm{u}(\mathrm{p}_{u_{n_s^*}})$ re-XOR $\mathrm{p}_{u_{n_s^*}}$ with $\mathcal{H}_{u_k}$.
      – Calculate the completion time of each
        $u_k \in \mathrm{u}(\mathrm{p}_{u_{n_s^*}})$ as in (2).
      – Each $u_k \in \mathrm{u}(\mathrm{p}_{u_{n_s^*}})$ broadcasts one bit ACK to all
        players in $\mathcal{Z}_{u_k}$.
  • End for
  • $t = t + 1$.
  • Update $\mathcal{H}_{u_k}, \mathcal{W}_{u_k}, \forall u_k \in \mathcal{N}_w$.
Until $\mathcal{W}_{u_k} = \phi, \forall u_k \in \mathcal{N}_w$.;
Output: Transmission slot $t$.
```

some packets to each visited coalition in previous transmissions.

- The connectivity constraints of UDs are incorporated in the coalition formations by: i) forming disjoint coalitions of nearby UDs based on their preferences, and ii) selecting the potential transmitting UD that can only target its connected players in its coalition. As such, we make sure that connected UDs are grouped into a single coalition.

In the considered game, each player has two actions to take such as, it can either transmit a packet combination p or it can listen to a transmission. Therefore, the action of the u_k-th player at the t-th transmission slot is $\mathcal{AC}_{u_k,t} = \{\text{transmit } \mathrm{p}_{u_k}, \text{ remain silent}\}$. The asymmetry of the side information at each player generates a different packet combination to be sent by each player at each transmission slot. This causes the asymmetry of the action space of each player. Also, in each transmission, different players are associated with each coalition. All these make the payoff of each coalition unique.

V. CONVERGENCE ANALYSIS, COMPLEXITY, AND COMMUNICATION OVERHEAD

A. CONVERGENCE AND STABILITY

The stability of the coalition structures in coalition formation games corresponds to an equilibrium state known as Nash-equilibrium. Here, we prove that the resulting coalition structure from Algorithm 1 is converged, and it is a Nash-stable.

The following theorem demonstrates that Algorithm 1 terminates in a finite number of iterations.

Theorem 1: Given any initial coalition structure Ψ_{ini}, the coalition formation step of Algorithm 1 maps to a sequence of merge-and-split rules which converges, in a finite number of iterations, to a final coalition structure Ψ_{fin} composed of a number of disjoint coalitions.

Proof: To proof this theorem, we need to show that for any merge or split rule, there exists a new coalition structure that results from the coalition formation step of Algorithm 1. Starting from any initial coalition structure Ψ_{ini}, the coalition formation step of Algorithm 1 can be mapped to a sequence of merge/split rules. As per definition 8 and definition 9, every merge or split rule transforms the current coalition structure into another coalition structure. Hence, we obtain the following sequence of coalition structures

$$\Psi_{\text{ini}} \rightarrow \Psi_1 \rightarrow \Psi_2 \rightarrow \cdots \Psi_{\text{fin}} \tag{9}$$

where $\Psi_{i+1} \triangleright \Psi_i$, and $\rightarrow$ indicates the occurrence of a merge-and-split rule. Since the Pareto Order introduced in definition 6 is irreflexive, transitive, and monotonic, a coalition structure cannot be revisited. Given the fact that the number of merge and split rules of a finite set is *finite*, the number of coalition structure sequences in (9) is finite. Therefore, the sequence in (9) always terminates and converge to a final coalition structure Ψ_{fin}. ∎

Definition 9: A coalition structure $\Psi = \{\mathcal{S}_1, \cdots \mathcal{S}_m\}$ *is a Nash-stable if players have no incentive to leave* Ψ *through merge-and-split operations.*

This definition implies that any coalition structure Ψ is considered as a Nash-stable coalition structure if and only if no player has an incentive to move from its current coalition and join another coalition or make an individual decision by performing any merge/split rules. Further, the players in the final coalition structure Ψ_{fin} have no incentive to do more merge and split operations. A Nash-stable coalition structure is also an individually stable coalition structure. In general, in a coalition formation game, Nash-stability is a subset of individual stability [47]. Specifically, no player leaves its current coalition through a split rule and form an empty coalition, i.e., no singleton coalition is created if the following property holds.

Property 2: There exists at least one coalition structure Ψ *that satisfies both Nash-stability and individual stability if and only if* $\forall \mathcal{S}_s \in \Psi$ *such that* $|\mathcal{S}_s| > 1$.

Proof: This property states that forming a singleton coalition cannot happen. Indeed, since each player cannot send an encoded packet to itself, it believes that a better payoff can be obtained by being a member of any coalition. Further, since the payoff of a non-targeted player in any coalition and a single player-coalition is the same, our proposed algorithm, as mentioned in the previous section, avoids making any merge-and-split rules for equal payoff values. Thus, according to Algorithm 1, a Nash-stable and individual stable coalition structure can be obtained. ∎

As a consequence of Property 2, the final coalition structure Ψ_{fin} that results from Algorithm 1 is $\mathbb{D}_{\text{hp}}$ stable as the coalitions have no incentive to do further merge-and-split operations. $\mathbb{D}_{\text{hp}}$ stable is also known as merge-and-split proof [47]. Furthermore, Ψ_{fin} can be considered as $\mathbb{D}_{\text{c}}$ stable. This is because players have no incentive to leave Ψ_{fin} and form any other coalitions [38]. In what follows, we provide an example to discuss the aforementioned stability concepts of the resulting coalition structure from Algorithm 1.

B. COALITION FORMATION EXAMPLE

In order to illustrate the proposed coalition formation algorithm and its stability concepts, consider the example presented in Fig. 2. For ease of analysis, we assume error-free transmissions. Given the coverage zone of the players and their side information as in Fig. 2, the resulting coalition structure Ψ_{fin} from Algorithm 1 consists of two disjoint coalitions $\{\mathcal{S}_1, \mathcal{S}_2\}$ where only one player transmits in each coalition. In particular, in coalition $\mathcal{S}_1$, player u_4 transmits packet p_1 to player u_6, and in coalition $\mathcal{S}_2$, player u_1 transmits the combination $p_3 \oplus p_4$ to players u_2, u_3, and u_5. The transmitting player in each coalition is shown in red circle, and their targeted players and the packet combinations are shown in Fig. 3. Given the resulting coalition structure $\Psi_{\text{fin}} = \{\mathcal{S}_1, \mathcal{S}_2\}$ that shown in Fig. 3, we now analyze its Nash stability. The coalition structure Ψ_{fin} is a Nash-stable as no player has an incentive to leave its current coalition. For example, player u_5 has a payoff of $\phi_{u_5}(\mathcal{S}_2) = -2$ when being part of the coalition $\mathcal{S}_2 = \{u_1, u_2, u_3, u_5\}$. The payoff $\phi_{u_5}(\mathcal{S}_2)$ is calculated as follows. Since player u_5 receives an instantly-decodable combination from player u_1, it does not experience any decoding delay increases. Thus, by (2), its anticipated completion time is $\mathcal{T}_{u_5}(\underline{n}_t) = \frac{|\mathcal{W}_{u_5,0}| + \mathbb{D}_{u_5}(\underline{n}_t) - \mathbb{E}[\sigma_{u_5}]}{1 - \mathbb{E}[\sigma_{u_5}]} = 2$, and, by (5) its payoff is -2. If player u_5 switches to act non-cooperatively and joins $\mathcal{S}_1$, player u_6 can be the new transmitting player in $\mathcal{S}_1$. In this case, player u_5 will be in the coverage zone of both transmitting players u_1 in $\mathcal{S}_2$ and u_6 in $\mathcal{S}_1$. Consequently, the payoff of player u_5 decreases to $\phi_{u_5}(\mathcal{S}_1) = -3$, and the payoff of player u_6 decreases from $\phi_{u_6}(\mathcal{S}_1) = -3$ to $\phi_{u_6}(\mathcal{S}_1) = -4$. Thus, player u_5 does not deviate form its current coalition $\mathcal{S}_2$ and join $\mathcal{S}_1$. Similarly, if players u_2 and u_3 act non-cooperatively by leaving $\mathcal{S}_2$ and forming a singleton coalition for each, i.e., $\mathcal{S}_3$ and $\mathcal{S}_4$, their payoffs decrease from $\phi_{u_2}(\{2\}) = -2$ and $\phi_{u_3}(\{3\}) = -1$ to $\phi_{u_2}(\mathcal{S}_3) = -3$ and $\phi_{u_3}(\mathcal{S}_4) = -2$, respectively. Clearly, Ψ_{fin} is an individual Nash-stable as it does not have any singleton coalition. Further, it is both $\mathbb{D}_{\text{hp}}$ and $\mathbb{D}_{\text{c}}$ stable as no further merge-and-split operations can be performed by the coalitions and no player has incentive to deviate from Ψ_{fin}, respectively.

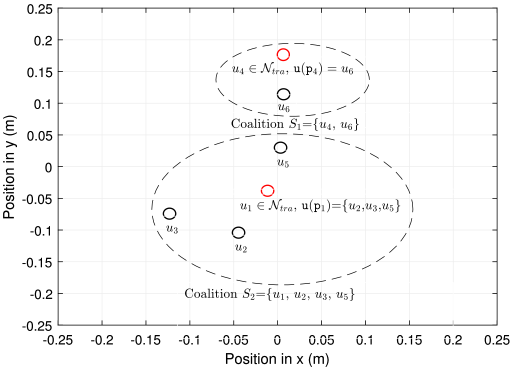

FIGURE 3. A resulting coalition structure $\Psi_{\text{fin}} = \{\mathcal{S}_1, \mathcal{S}_2\}$ from Algorithm 1 for a partially connected D2D network that is presented in Fig. 2.

C. COMPLEXITY ANALYSIS AND COMMUNICATION OVERHEAD

Here, we analyze the complexity and communication burden of Algorithm 1.

1) COMPLEXITY

Each UD at any transmission slot generates a packet combination that depends on its previously received packets. Further, since a game with incomplete information, i.e., each UD knows only the received and requested packets of UDs in its coverage zone, every UD generates its packet combinations of all other UDs in its coverage zone. This allows every UD to calculate the payoff function (5) of all other UDs in its coverage zone.

The complexity of generating the packet combination by the CP using a maximum weight search method is given as follows. First, the CP generates all the vertices that represent all UDs and their requested packets, which needs a computational complexity of $O(NM)$. Note that a vertex is generated for each file p_l that is being requested by each UD u_k, $\forall p_l \in \mathcal{P}, \forall u_k \in \mathcal{N}$. The CP, then, connects these vertices (circles) by edges (connections) that represent network coding conditions. This needs a computational complexity of $O(N^2M^2)$. Afterward, the CP executes the maximum weight search method that computes the weight of $O(NM)$ vertices and selects a maximum of N UDs. Hence, the overall complexity of finding the IDNC packet is $O(NM) + O(N^2M^2) + O(N^2M) = O(N^2M^2)$ operations [11]. In our case, the computational complexity for generating the packet combination at each UD is bounded by $O(N^2M^2)$. This is because the number of UDs in the coverage zone of each UD is less than the total number of UDs in the network.

2) COMMUNICATION OVERHEAD

The communication overhead of Algorithm 1 is related to perform the members' discovery step, transmitting player selection, and the analysis of merge-and-split rules. The communication overhead analysis is explained as follows.

Step 1: Similar to many algorithms in the literature, e.g., [48], the member discovery step needs $|N|$ 2-byte messages, in which each message is being sent by every UD to all its neighbors which is denoted by $\mathbf{N}$. Thus, the total communication overhead for discovering the neighbor UDs is $|2N\mathbf{N}|$ bytes.

Step 2: Selecting the transmitting player in each coalition can be performed in many different strategies, e.g., based on players' attributes [49], [50]. In Algorithm 1, coalition's members initially exchange an advertisement message among them, and the one that satisfies the conditions in Section IV-B can be selected as a transmitting player in that coalition. The same process is applied for selecting/updating the transmitting player in step III. Since it is connected to most players in the coalition, the transmitting player is responsible for ensuring that the rest of the coalition's members received an acknowledgment (ACK). As such, they update their *Has* and *Wants* sets after each D2D transmission.

Step 3: The communication overhead due to forming coalitions is based on the number of merge-and-split operations. This is mainly related to the total number of decisions made by each of the N players. As previously mentioned in Section IV-B, the merge-and-split operations enumerate only the neighboring coalitions $\mathbb{S}_s$. Thus, two extreme cases can occur:

- If all coalitions' alliances decide to leave their current coalitions and join other coalitions. In this case, the u_k-th player in the $\mathcal{S}_s$-th coalition will make $|\mathbb{S}_s|$ decisions (the u_k-th player has $|\mathbb{S}_s|$ possibilities to join any of the neighbor coalitions). Consequently, the total number of players' decisions is $Q_{\text{worst}} = N|\mathbb{S}_s|$.
- If players did not make any decisions. Since players make no decision, the overhead, in this case, is only $Q_{\text{best}} = N$ (due to the initial player-coalition associations in step 1).

In practical, the number of players' decisions is between the above two cases, i.e., $Q_{\text{best}} \leq Q \leq Q_{\text{worst}}$. Hence, if L average decisions are made by players, then $Q = NL$ decisions that perform split-and-merge rules in Algorithm 1.

Therefore, combining all the overhead signaling components, the total overhead in each transmission is $N(2\mathbf{N} + L)$. Such signaling cost will add only a few bytes, which are negligible in size compared to the entire packet's size. Furthermore, to update the *Has* and *Wants* sets of UDs, only the indices of packets need to be exchanged between the UDs, not their contents. Hence, we ignore the overhead signaling factor because it is a constant (independent on the completion time and decoding delay) and its size is negligible.

VI. NUMERICAL RESULTS

This section evaluates the performance of our proposed coalition formation game (denoted by CFG partially-connected D2D) to demonstrate its capability of reducing the completion time compared to existing schemes. We first give the simulation setup and implemented schemes. Then, the completion time and game performances are comprehensively evaluated.

A. SIMULATION SETUP

We consider a partially connected D2D network where every UD is connected to some other UDs within its coverage zone based on the connectivity index C, which is defined as the ratio of the average number of neighboring UDs to the total number of UDs N. The considered model is shown in Fig. 1 in the system model section. Further, the UDs in the considered model are uniformly re-positioned after each transmission in a 500m×500m cell. The system setting in this paper follows the setup studied in [28], [29]. As mentioned in the system model, each UD has already acquired some packets and wants some other packets from the set $\mathcal{P}$. Such initial *Has* and *Wants* sets $\mathcal{H}_{u_k}$ and $\mathcal{W}_{u_k}$, $\forall u_k \in \mathcal{N}$ of UDs is independently drawn based on their erasure probabilities. The short-range D2D links are more reliable than the CP-UD communications [26], [27]. Hence, we assume that the player-to-player erasure probability σ is half the CP-to-UD erasure ϵ, i.e., $\sigma = 0.5\epsilon$. Unless otherwise stated, the simulation parameters are listed in Table 2. Our simulations were implemented using Matlab on a Windows 10 laptop 2.5 GHz Intel Core i7 processor and 8 GB 1600 MHz DDR3 RAM. For the sake of comparison, we implement the following schemes.

- **The fully-connected D2D scheme:** In this scheme, a single UD who has the largest number of received

TABLE 2. Simulation parameters.

Cellular layout	Square with 500m×500m
Channel model	Packet erasure channel
Distribution of UDs	Uniformly distributed
UDs' side information	Randomly distributed
Connectivity index C	0.4
UD-to-UD erasure probability σ	0.12
CP-to-UD erasure probability ϵ	0.25

packets transmits an IDNC packet at each transmission slot.

- **The Point to Multi-Point (PMP) scheme:** The CP in this scheme is responsible for the transmissions. It provides all the requested packets to all the UDs. This scheme was proposed in [15].
- **The one coalition formation game:** This scheme is denoted by OCF partially-connected D2D. Only one coalition is formed in this scheme, and accordingly, a single UD transmits an IDNC packet at each transmission. The transmitting UD is selected based on its number of received packets as well as on the maximum number of UDs in its coverage zone.
- **The FRAN partially-connected D2D scheme:**. In this scheme, a fog central unit is responsible for determining the set of transmitting UDs and their corresponding packet combinations. The content delivery problem in partially-connected D2D network was solved by optimization as explained in [29].

B. COMPLETION TIME PERFORMANCE ANALYSIS

The total number of transmissions, that are required until all UDs receive all their requested packets, measure the overall completion time. Thus, all the simulated schemes are executed until all UDs receive all their requested packets. The presented average values of the number of transmissions are computed over a certain number of iterations. To study the number of required transmissions of the proposed solution, we change the number of players, packets, connectivity index, and packet's erasure probability.

In Fig. 4, we depict the average number of transmissions versus the number of players N for a network composed of $M = 30$ packets, $\epsilon = 0.25$, $\sigma = 0.12$, and connectivity index $C = 0.4$. It is observed from Fig. 4 that the proposed CFG partially-connected D2D algorithm outperforms the PMP, fully-connected D2D, and OCF partially-connected D2D schemes for all simulated numbers of players. This is because of the simultaneous IDNC packet transmissions from cooperating players. In particular, the fully-connected D2D system only considers the size of the *Has* set as a metric to select a single UD for transmission at each transmission slot, i.e., $u_{n^*} = \max\limits_{u_n \in \mathcal{N}} \mathcal{H}_{u_n}$. The OCF partially-connected D2D scheme focuses on the maximum number of connected UDs to be formed as well as on the size of the *Has* set of the transmitting UD. On the other hand, although the transmitter in the PMP scheme can encode all the IDNC combinations and schedule a certain number of UDs, the PMP scheme sacrifices the utility of the simultaneous transmissions by considering only one transmission. Our proposed algorithm strikes a balance between the aforementioned aspects by jointly considering the number of scheduled UDs, their erasure probabilities, and the *Has* set size of each transmitting UD. Despite the gain achieved by the partially-connected D2D fog solution with the presence of a fog that executes the whole process, our decentralized solution offers almost the same completion time performance. This is due to the fact that the centralized D2D fog and our proposed schemes consider the same philosophy of D2D simultaneous transmissions.

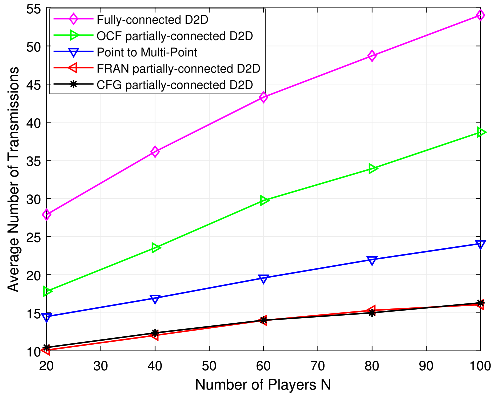

FIGURE 4. Average number of transmissions versus the number of players *N*.

We observe from Fig. 4 that, for a small number of players, the completion time performance of the PMP system is close to both the CFG partially-connected D2D and FRAN partially-connected D2D schemes. This is because, for a few number of players ($N \leq 60$), the certainty that the whole frame M is distributed between players in the initial transmissions is low, thus decreasing the probability of delivering potential IDNC packets to UDs. This makes the overall completion time performance of the partial D2D scenarios close to the PMP scheme. As the number of players increases ($N \geq 80$), the bigger the certainty that the union of their *Has* sets is equal to M. This results in delivering more potential D2D IDNC packets, thus increasing the gap between the PMP performance and both the FRAN partially-connected D2D and proposed schemes.

In Fig. 5, we illustrate the average number of transmissions versus the number of packets M for a network composed of $N = 30$ UDs, $\epsilon = 0.25$, $\sigma = 0.12$, and connectivity index $C = 0.4$. The figure shows that the proposed scheme outperforms the fully connected, one coalition game, and PMP schemes. For a few packets, the IDNC combinations are limited, which affects the ability of the proposed scheme to generate coded packets that satisfy many scheduled UDs. With increasing the number of packets, the number of transmissions needed for the completion of the aforementioned schemes is remarkably increasing. Therefore, as the number of packets increases, the proposed scheme substantially

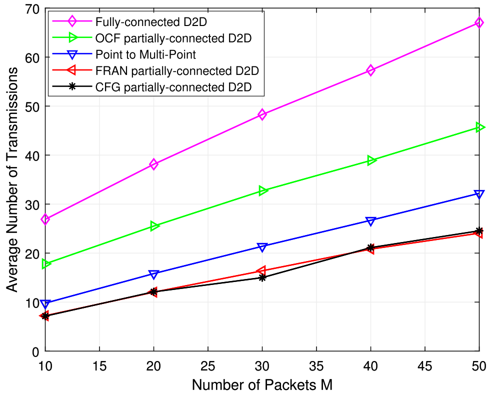

FIGURE 5. **Average number of transmissions versus the number of packets *M*.**

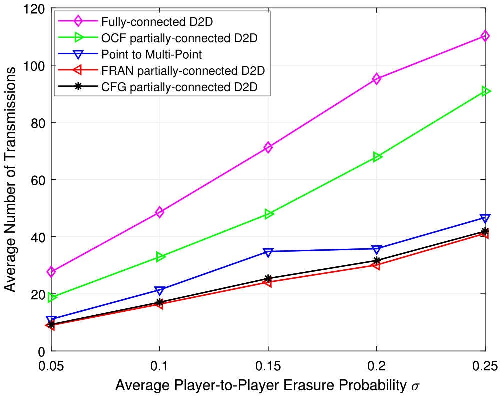

FIGURE 6. **Average number of transmissions versus the average player-player erasure probability σ.**

outperforms the fully connected and one coalition game schemes. We see from Fig. 4 that the number of transmissions of all schemes linearly increases with the number of packets. This is expected as the number of packets increases, a high number of transmissions is required for delivering all packets to all UDs. This results in increasing the overall completion time.

In Fig. 6, we plot the average number of transmissions versus the average player-player erasure probability σ for a network composed of $N = 60$, $M = 30$, $\epsilon = 2\sigma$, and $C = 0.4$. Similar to what we have discussed in Figs. 4 and 5, the average number of transmissions of the partial D2D schemes is noticeable compared to the fully-connected D2D and OCF partially-connected D2D schemes, as shown in Fig. 6. We clearly see that the number of required transmissions of the partial D2D schemes is less than the number of transmissions of the PMP scheme because of their multiple UDs' transmissions at each transmission slot. Moreover, as the player-to-player erasure probability increases, the CP-UD erasure probability increases two-fold ($\epsilon = 2\sigma$), thus relatively affecting the completion time performance of the PMP scheme. The partial D2D settings, however, benefit from short D2D links, which provide much better UDs reachability and packet successful delivery compared to the PMP setting.

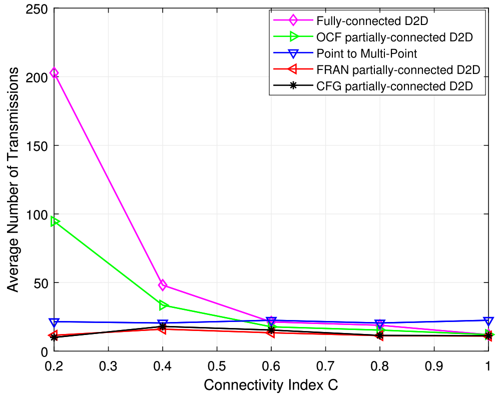

FIGURE 7. **Average number of transmissions versus the connectivity index *C*.**

In Fig. 7, we investigate the average number of transmissions versus the connectivity index C for a network composed of $N = 60$, $M = 30$, $\epsilon = 0.25$, and $\sigma = 0.12$. It can clearly be seen that for a low connectivity index ($C \leq 0.4$), the proposed CFG partially-connected D2D scheme noticeably outperforms the fully-connected D2D and OCF partially-connected D2D schemes. In such poorly connected networks ($C \leq 0.4$), multiple simultaneous players' transmissions are exploited in partially D2D algorithms. However, as the connectivity index increases ($C \geq 0.6$), the number of formed disjoint coalitions in our proposed solution is drastically reduced, thus reducing the number of transmitting UDs. This results in a performance agreement with the fully-connected D2D scheme. Being independent of the coverage zones of the transmitting UDs and the delay created by those UDs, the PMP scheme is not affected by the changes to C. Thus, the PMP scheme has a constant average number of transmissions.

To conclude this section, we study the influence of the setting $\sigma = 0.5\epsilon$ on the number of transmissions of our proposed scheme. In Table 3, we present the average number of transmissions for different values of σ. The considered network setup has 30 UDs, 20 packets, $\epsilon = 0.5$, and $C = 0.1$. From Table 3, we notice that the performance of our proposed solution still outperforms the PMP scheme for $\sigma = 0.7\epsilon$ and approximately offers the same performance as for the PMP scheme for $\sigma = 0.9\epsilon$. This is due to the simultaneous transmissions and cooperative decisions by the transmitting UDs, which show the potential of the proposed CFG solution in minimizing the number of required transmissions for packet delivery.

C. PROPOSED CFG PERFORMANCE ANALYSIS

To quantify the analysis of the proposed formation coalition solution, we plot in Fig. 8 the average number of coalitions versus the number of players N for a network composed of $M = 30$, a different connectivity index ($C = 0.6$, $C = 0.3$, and $C = 0.1$), and $\sigma = 0.12$. Fig. 8 shows that the average coalition size increases with the increase in the number of players. This is because, as N increases, the number of

TABLE 3. The influence of changing σ on the average number of transmissions of the proposed scheme.

Solution	$\sigma = 0.6\epsilon$	$\sigma = 0.7\epsilon$	$\sigma = 0.9\epsilon$	$\sigma = \epsilon$
Point to Multi-Point	30.2900	30.2800	30.3100	30.4800
CFG partially-connected D2D	20.1800	23.4702	30.4500	33.9300

TABLE 4. Average run times of the different schemes.

Solution	Time(s)- Small network	Time(s)- Large network
FRAN partially-connected D2D	0.561893	15.98450
Point to Multi-Point	1.994500	1103.020716
Fully-connected D2D	0.756420	128.772580
OCF partially-connected D2D	0.783575	28.726515
CFG partially-connected D2D	0.736737	21.725739

TABLE 5. Average number of coalitions and split/merge rules of the proposed scheme in the first iteration.

Network Setup	Number of Coalitions	Split-and-merge rules
Setup 1: $N = 100$ and $C = 0.1$	16.34	8.12
Setup 2: $N = 160$ and $C = 0.1$	23.67	12.76

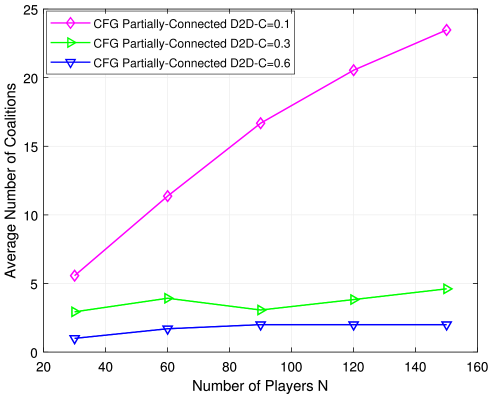

FIGURE 8. Average number of coalitions versus the number of players ***N***.

cooperating players increases, thus increasing the average size of the formed coalitions. We can conclude from Fig. 8 that the resulting coalition structure Ψ_{fin} from Algorithm 1 is composed of a small number of relatively large coalitions when $C = 0.6$. When $C = 0.1$, this number of formed coalitions increases, and the resulting coalition structure Ψ_{fin} is composed of a large number of small coalitions' sizes.

In Table 4, we evaluate the complexity of the proposed coalition game solution as a function of the run time. In particular, Table 4 lists the consumed time of MATLAB to execute all schemes in different network setups since starting the algorithms until all players receive their wanted packets. The considered small network setup has 30 players, 20 packets, $\epsilon = 0.5$, $\sigma = 0.25$, and $C = 0.1$. The considered large network setup has 100 players, 70 packets, $\epsilon = 0.5$, $\sigma = 0.25$, and $C = 0.1$. It can clearly be seen from the table that the proposed CFG-partially D2D scheme needs low consumed time than all other solutions for both network setups. Although the completion time achieved by the CFG partially-connected D2D scheme is roughly the same as the centralized FRAN partially-connected D2D, the computing time required by our developed scheme is slightly higher than that required by the FRAN partially-connected D2D. This is because our proposed scheme needs time to converge before generating the output. The centralized FRAN scheme has low run time due to the presence of the fog unit.

To evaluate the convergence rate analysis of the proposed scheme, the average number of merge-and-split rules before Algorithm 1 converges to the final coalition structure is listed in Table 5. To achieve the stable coalition with our proposed CFG scheme, network setup 1 requires, on average 16 iterations, and network setup 2 needs, on average 22 iterations. These results show that our proposed distributed algorithm is robust to different network setups. In summary, these results show that our proposed algorithm allows UDs to quickly form converged and stable coalition structures, which confirms the theoretical findings in Theorem 1.

Finally, some advantageous of our proposed scheme compared to the one in [29] are given as follows. First, our distributed and decentralized scheme offers almost the same completion time performance of the centralized work in [29] while offloading the fog unit. Such offloading policy will allow the fog to serve more UDs. Second, our scheme alleviates the required high power at UDs to deliver their channels' status to the fog by using reliable and short D2D links. Third, if some UDs move from one fog to another, their lost and received content status have to be transferred to the new fog. This requires implementing handover procedure at the fogs which is not an issue in our distributed scheme as each UD receives its requested packets from other nearby UDs.

VII. CONCLUSION

In this paper, we have developed a distributed game-theoretical framework for a partially connected D2D network using coalition game and IDNC optimization. As such, the distribution of contents among UDs is minimized. In particular, our proposed model is formulated as a coalition formation game with non-transferable utility, and a fully distributed coalition formation algorithm is proposed. The proposed distributed algorithm is converged to a Nash-stable coalition structure using split-and-merge rules while accounting for the altruistic players' preferences. With such a distributed solution, each UD has to maintain a partial feedback matrix only for the UDs in its coverage zone instead of the global feedback matrix required in the fully connected D2D networks. A comprehensive completion time and game performance evaluations have been carried out for the proposed distributed solution. In particular, our performance evaluation results comprehensively demonstrated that the proposed distributed solution offers almost the same completion time performance similar to the centralized FRAN D2D network.

REFERENCES

[1] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, 4th Quart., 2014.

[2] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Commun.*, vol. 19, no. 3, pp. 96–104, Jun. 2012.

[3] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[5] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*, Cisco, San Jose, CA, USA, Feb. 2019.

[6] T. Wang, Y. Sun, L. Song, and Z. Han, "Social data offloading in D2D-enhanced cellular networks by network formation games," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 7004–7015, Dec. 2015.

[7] C. Xu, C. Gao, Z. Zhou, Z. Chang, and Y. Jia, "Social network-based content delivery in device-to-device underlay cellular networks using matching theory," *IEEE Access*, vol. 5, pp. 924–937, Nov. 2017.

[8] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[9] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[10] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1–6.

[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] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. IEEE GLOBECOM*, Miami, FL, USA, Dec. 2010, pp. 1–5.

[13] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, no. 1, pp. 1–14, Dec. 2010.

[14] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

[15] 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.*, Dec. 2014, pp. 5008–5013.

[16] A. Douik, M. S. Al-Abiad, and M. J. Hossain, "An improved weight design for unwanted packets in multicast instantly decodable network coding," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 2122–2125, Nov. 2019.

[17] L. Lu, M. Xiao, and L. K. Rasmussen, "Design and analysis of relay-aided broadcast using binary network codes," *J. Commun.*, vol. 6, no. 8, pp. 610–617, Nov. 2011.

[18] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. IEEE ISIT*, Seoul, South Korea, Jun. 2009, pp. 844–848.

[19] 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 J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, May 2011.

[20] M. S. Al-Abiad, A. Douik, and S. Sorour, "Rate aware network codes for cloud radio access networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1898–1910, Aug. 2019.

[21] M. S. Al-Abiad, M. J. Hossain, and S. Sorour, "Cross-layer cloud offloading with quality of service guarantees in fog-RANs," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8435–8449, Dec. 2019.

[22] M. S. Karim, P. Sadeghi, S. Sorour, and N. Aboutorab, "Instantly decodable network coding for real-time scalable video broadcast over wireless networks," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, pp. 1–24, Jan. 2016.

[23] M. S. Al-Abiad, A. Douik, S. Sorour, and M. D. J. Hossain, "Throughput maximization in cloud radio access networks using network coding," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas, MO, USA, May 2018, pp. 1–6.

[24] M. S. Al-Abiad, S. Sorour, and M. J. Hossain, "Cloud offloading with QoS provisioning using cross-layer network coding," in *Proc. IEEE Globecom*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[25] M. Saif, A. Douik, and S. Sorour, "Rate aware network codes for coordinated multi base-station networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–7.

[26] N. Aboutorab and P. Sadeghi, "Instantly decodable network coding for completion time or decoding delay reduction in cooperative data exchange systems," *IEEE Trans. Veh. Tech.*, vol. 65, no. 3, pp. 1212–1228, Feb. 2015.

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

[28] 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," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 7040–7053, Oct. 2018.

[29] 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.

[30] 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.

[31] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[32] M. Chen, W. Saad, and C. Yin, "Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[33] A. Douik, S. Sorour, H. Tembine, T. Y. Al-Naffouri, and M.-S. Alouini, "A game-theoretic framework for decentralized cooperative data exchange using network coding," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 901–917, Apr. 2017.

[34] 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. IEEE Global Commun. Conf.*, Austin, TX, USA, Jun. 2014, pp. 1583–1589. Apr. 2017.

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

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

[37] 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.

[38] K. Apt and A. Witzel, "A generic approach to coalition formation (extended version)," in *Int. Game Theory Rev.*, vol. 11, no. 3, pp. 347–367, Mar. 2009.

[39] L. Militano, A. Orsino, G. Araniti, A. Molinaro, and A. Iera, "A constrained coalition formation game for multihop D2D content uploading," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2012–2024, Mar. 2016.

[40] C. Xu, J. Feng, Z. Zhou, J. Wu, and C. Perera, "Cross-layer optimization for cooperative content distribution in multihop device-to-device networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 278–287, Feb. 2019.

[41] Y. Zhao, Y. Li, Z. Ding, N. Ge, and H. V. Poor, "A coalitional graph game framework for network coding-aided D2D communication," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, p. 2, Dec. 2016.

[42] C. Gao, Y. Li, Y. Zhao, and S. Chen, "A two-level game theory approach for joint relay selection and resource allocation in network coding assisted D2D communications," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2697–2711, Oct. 2017.

[43] A. Antonopoulos and C. Verikoukis, "Multi-player game theoretic MAC strategies for energy efficient data dissemination," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 592–603, Feb. 2014.

[44] A. Antonopoulos, J. Bastos, and C. Verikoukis, "Analogue network coding-aided game theoretic medium access control protocol for energy-efficient data dissemination," *IET Sci., Meas. Technol.*, vol. 8, no. 6, pp. 399–407, Nov. 2014.

[45] M. Zayene, O. Habachi, V. Meghdadi, T. Ezzedine, and J. P. Cances, "A coalitional game-theoretic framework for cooperative data exchange using instantly decodable network coding," *IEEE Access*, vol. 7, pp. 26752–26765, 2019.

[46] P. Sadeghi, R. Kennedy, P. Rapajic, and R. Shams, "Finite-state Markov modeling of fading channels–a survey of principles and applications," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 57–80, Sep. 2008.

[47] K. Apt and T. Radzik, "Stable partitions in coalitional games," May 2006, *arXiv:cs/0605132*. [Online]. Available: https://arxiv.org/abs/cs/0605132

[48] Z. Han and K. J. Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[49] B. P. Deosarkar, N. S. Yadav, and R. P. Yadav, "Clusterhead selection in clustering algorithms for wireless sensor networks: A survey," in *Proc. Int. Conf. Comput., Commun. Netw.*, Dec. 2008, pp. 1–8.

[50] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.

MOHAMMED S. AL-ABIAD (Student Member, IEEE) received the B.Sc. degree in computer and telecommunications engineering from Taiz University, Taiz, Yemen, in 2010, and the M.Sc. degree in telecommunication engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2017. He is currently pursuing the Ph.D. degree with The University of British Columbia, Canada. His research interests include cross-layer network coding, cloud-radio access networks, device-to-device communications, caching, and game theory.

AHMED DOUIK (Member, IEEE) received the Engineering degree (Hons.) in electronic and communication engineering from the Ecole Polytechnique de Tunisie, Tunisia, in 2013, the M.S. degree in electrical engineering from the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, in 2015, and the Ph.D. degree in EE from the California Institute of Technology, Pasadena, CA, USA, in 2020. His research interests include convex and non-convex optimization, machine learning, signal processing, and communications.

MD. JAHANGIR HOSSAIN (Senior Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from the Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, the M.A.Sc. degree from the University of Victoria, Victoria, BC, Canada, and the Ph.D. degree from The University of British Columbia (UBC), Vancouver, BC. He was a Lecturer with BUET. He was a Research Fellow with McGill University, Montreal, QC, Canada, the National Institute of Scientific Research, Quebec, QC, and the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA, Australia. His industrial experience includes a Senior Systems Engineer with Redline Communications, Markham, ON, Canada, and a Research Intern with Communication Technology Laboratory, Intel Inc., Hillsboro, OR, USA. He is currently an Associate Professor with the School of Engineering, UBC, Canada. His research interests include designing spectrally and power-efficient modulation schemes, applications of machine learning for communications, quality-of-service issues and resource allocation in wireless networks, and optical wireless communications. He regularly serves as a member of the Technical Program Committee of the IEEE International Conference on Communications (ICC) and the IEEE Global Telecommunications Conference (Globecom). He has served as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has served as an Associate Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has been serving as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.