

A Proactive Scalable Approach for Reliable Cluster Formation in Wireless Networks with D2D Offloading

Sanaa Sharafeddine and Omar Farhat
 $\{sanaa.sharafeddine, omar.farhat\}@lau.edu.lb$
 Lebanese American University, Beirut, Lebanon

Abstract—With the current exponential growth in traffic and service demands, device-to-device (D2D) cooperation is identified as a major mechanism to enable 5G networks to effectively and efficiently augment network resources. The effectiveness of D2D cooperation depends on a wide range of decision making processes that include cluster formation, resource allocation, in addition to connection and mobility management. Irrespective of the D2D cooperation scenario whether in sensor, ad hoc, or cellular networks, the literature normally assumes that devices selected as relays or data sources are reliable; this means that they will maintain the connection till the communication session ends. Yet, this assumption is challenged in practice since devices' batteries can be depleted (e.g., sensors in an IoT network) and devices can move leading to connection termination (e.g., mobile users in a WiFi network or cars in a vehicular ad hoc network). To this end, we address the problem of reliable D2D cooperation in wireless networks by proposing a novel approach that is proactive by utilizing reliability metrics in the decision making process, and scalable by having low implementation complexity suitable for dense networks. These differentiating factors are shown to enhance the overall network reliability compared to standard techniques and to facilitate dynamic operation which is essential for practical implementation. Performance is evaluated using extensive simulations in addition to test bed experimental demonstration in order to quantify gains and extract insights on a range of existing design tradeoffs.

Index Terms – Device-to-device cooperation, mobile-to-mobile data sharing, traffic offloading, robust clustering, wireless test bed design, reliable communications

I. INTRODUCTION

Device-to-device (D2D) communication is expected to play a key role in 5G systems to provision ultra dense networks with improved performance, reduced latency, and lower energy consumption. Forming reliable D2D communications is yet another challenge that is foreseen to realize ultra-reliable low latency communications. Cooperation among devices has been shown to be highly effective in enhancing the performance of infrastructure-based wireless networks for a wide range of use cases; devices in the context of this work can be sensor nodes in an Internet of Things (IoT) network, mobile users in a cellular or WiFi network, or cars in a vehicular ad hoc network (VANET) [1], [2], [3], [4], [5]. Facilitating mobile-to-mobile, sensor-to-sensor or car-to-car (all denoted as device-to-device (D2D) in the sequel) cooperation can lead to traffic offloading, throughput enhancement, energy consumption savings, coverage extension, and/or cost reduction, e.g., see [6], [7], [8], [9] and the references therein.

A major challenge that hinders reaping the benefits of cooperation in wireless networks is reliability and robustness

against device-level dynamics as channel conditions vary, devices move, and devices' data traffic and energy/processing capabilities change over time. In infrastructure-based cooperative wireless networks, an essential element is the grouping of nodes into clusters whereby cluster head devices are selected intelligently to act as relays between infrastructure nodes (e.g., cellular base stations or WiFi access points) and other devices; the data communication within any cluster can be disrupted upon the loss of its cluster head device, due to battery drain, device mobility, device malfunctioning, or even a malicious security attack. To this end, in this work we focus on the design of a proactive scalable approach for reliable cluster formation in cooperative wireless networks while achieving target performance requirements and having low complexity to be practically applicable to high dense scenarios.

A. Related Work

Even though there is rich literature on clustering techniques and solutions for cooperative wireless networks (e.g., see the surveys [10], [11] and the references therein), there is relatively less progress on addressing challenges related to fault tolerance and reliability. Failure recovery and survivability have been addressed in the context of wireless sensor and ad hoc network scenarios (e.g., see [12], [13], [14]). In sensor networks, failure recovery typically aims at maintaining connectivity as sensor devices die due to energy depletion, whereas in ad hoc networks a key aim is to adapt multihop routing in order to maintain network level connectivity as nodes move around. The related literature can in general be divided into reactive (post clustering) and proactive (pre clustering) reliability enhancement techniques.

Reactive techniques aim at either minimizing failures or recovering from failure events in a dynamic fashion after cluster formation. The authors in [15] present an RSSI (received signal strength indicator) based approach for cluster formation in sensor networks whereby cluster head selection takes into account signal strength level and energy budget among neighbor nodes with the option of dynamically replacing cluster heads based on given performance metrics. The authors in [16] present a mechanism for dynamic cluster head re-election by executing an update algorithm on a periodic basis using pre-configured parameters. The authors in [17] address fault tolerance in wireless sensor network scenarios using a distributed real-time recovery algorithm based on periodic protocol exchange messages to identify failures and deal with them. In [18], the authors presented several low

complexity reactive algorithms to deal with mobility in content distribution networks with D2D cooperation while taking into account three classes of events: an existing device leaves the network which is highly challenging when the device is a cluster head, an existing device moves locally within the network, and a new incoming device joins the network.

On the other hand, proactive techniques take preventive measures as part of the cluster formation process, either by considering device reliability metrics when electing cluster heads (also called group owners) or by identifying and assigning backup cluster heads to deal with failure events. For example, the authors in [19] present an interesting approach for group re-formation in WiFi-Direct D2D networks that includes the election of an emergency group owner serving as a backup cluster head and the configuration of dormant backend links for fast group establishment after failure events. The authors in [20], [21], [22], [23] present robust clustering algorithms for different network scenarios based on backup (also called secondary or redundant) cluster head selection taking into account metrics such as node degree and energy consumption. A hierarchical fuzzy logic based approach was developed in [24] whereby cluster heads connect to a super cluster head node acting as relay to a mobile base station. Short range D2D wireless technologies such as WiFi-Direct and Bluetooth do not include intelligence for group owner (cluster head) election as part of their standards. This triggered research to devise optimized mechanisms for cluster head election, either based on rotation among devices to distribute load over time such as the LEACH (low-energy adaptive clustering hierarchy) protocol [25] or based on biased selection taking into account performance metrics such as centrality of location or availability of energy budget [26], [27], [15], [28]. For example, the LEACH protocol divides time into rounds whereby in each round a new random sensor node is assigned as cluster head to evenly distribute the energy load among all sensors over time.

B. Contributions

In this paper, we present a novel proactive approach for reliable cluster formation among devices in infrastructure-based wireless network with D2D cooperation. The key differentiating factors of the proposed approach are its flexibility and scalability. The approach is flexible in proactively capturing different reliability metrics as part of a generic cost function and is flexible in its applicability to different wireless network scenarios including sensors, IoT devices, and mobile users. On the other hand, the approach is scalable to network scenarios with high density of devices as it has a relatively fast execution time. It is important to highlight that the proposed proactive approach can complement a wide range of existing state-of-the-art reactive techniques to further enhance the level of reliability. Thus, our proactive approach runs before initiating a given service to decide on cluster heads that will be managing D2D communications. During service operation and in the case of an unexpected cluster head failure, a complementary reactive approach has to be implemented. The reader is referred to Section I-A for a list of existing reactive approaches that can be incorporated.

In terms of solution methodology, we formulate the problem as an integer linear program and generate optimal results for small scale scenarios. For large scale scenarios, we utilize a fast and effective algorithm based on the notion of electrostatic forces; we demonstrate its effectiveness in achieving close-to-optimal results and its scalability by generating results for high dense environments with multiple access points. Finally, we extend the contributions to experimental evaluation using test bed implementation in order to demonstrate the algorithm's feasibility and effectiveness under realistic operational conditions. It is important to note that very few studies in the literature include practical test bed implementation due to the challenges in integrating intelligence into devices especially when dealing with off the shelf smartphones; for example, see [29], [30] for WiFi-Direct ad hoc network formation among Android phones, however, without dealing with intelligence related to cluster head election. Moreover, the authors in [16] present Android based implementation to demonstrate the practical implementation of their group creation and inter-communication approach using a network scenario composed of five devices.

In brief, the contributions of this work are multi-fold and can be summarized as follows: 1) A novel cluster formation mechanism in wireless networks with device-to-device cooperation is proposed to optimize network reliability; 2) A heuristic approach with close-to-optimal performance and high efficiency is proposed for networks with high density of devices such as IoT and WSN scenarios; 3) An experimental testbed is developed to demonstrate the practicality and effectiveness of the proposed algorithms in real scenarios.

C. Organization

Section II presents the system model including key assumptions and metrics. Section III includes the optimization problem formulation with explanation of the objective function, constraints, and problem complexity. Section IV presents the solution methodology including the details of the utilized algorithm; results and analysis are then summarized in Section V for a wide range of scenarios with various network and design parameters, including experimental evaluation. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL

We consider a wireless network scenario composed of N devices (sensors, IoT nodes, or mobile users) and M WiFi access points (APs) or cellular base stations in a given geographic area. The devices are equipped with multiple wireless interfaces supporting long range (LR) connectivity to the APs and short range (SR) connectivity to other device over direct D2D links (e.g., WiFi-Direct, LTE-Direct, or Bluetooth). It is worth noting here that such setup may lead to notable energy savings since communications within the cluster take place over short range connections that are more energy efficient than long range links [9], [31], [32]. Moreover, we assume centralized intelligence in the network side to facilitate D2D cooperation whereby the devices are divided into clusters with one device per cluster selected as a cluster head. The cluster

head acts then as a relay for data download between the APs and other devices in its cluster. Even though this is a standard system model, the novelty and differentiating factor of our work is the focus on reliability in the cluster formation process in order to minimize failure costs and enhance overall network performance. Fig. 1 presents a general schematic of the adopted system model and highlights that each device has certain characteristics reflected via battery status, availability of multiple interface, etc.

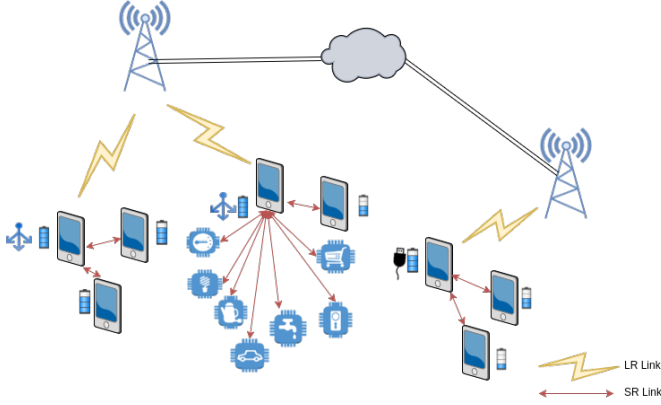


Fig. 1: Example system model.

We denote the reliability of a given device i as Γ_i , where Γ_i is a value between 0 and 1 and is a function of multiple factors including battery life denoted as E_i and device rating denoted as γ_i . For the energy metric, we define a percentage threshold β below which the device cannot cooperate due to limited budget and, thus, is marked non-reliable. On the other hand, for the device rating γ_i , we assume it is a value between 0 and 1 based on learning from historical data on the device's performance with respect to reliable cooperation, for example, based on the device's mobility patterns or D2D data sharing success rate. Connection failure due to mobility is a minor concern in limited mobility environments such as static WSN or IoT. In the case of more dynamic environments, devices with unexpected mobility pattern that disturbs existing links among neighboring devices would receive a lower γ_i to avoid selecting them as cluster heads in future D2D communications. We use the following expression to model the device's reliability, which gets non-zero value as long as γ_i is non-zero and the actual battery level is above the threshold βE where E represents the full battery capacity; the $E - \beta E$ factor in the denominator is used for normalization purposes to constrain Γ_i between 0 and 1:

$$\Gamma_i = \gamma_i \min \left(\frac{E_i - \beta E}{E - \beta E}, 0 \right) \quad (1)$$

Let $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$ denote the set of devices and $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$ denote the set of available access points. The system can then be modeled as a graph $G = (V, W)$, where $V = \mathcal{D} \cup \mathcal{S}$ and W the set of edges containing edge (u, v) if and only if two nodes u and v are within radio range with respect to each other. Moreover, we define the set $\mathcal{C} \subset \mathcal{D}$ composed of cluster head devices selected to relay downloaded data to other devices in the set $\mathcal{D} - \mathcal{C}$ using SR

TABLE I: Notations used in problem formulation

Notation	Description
N	Number of devices
M	Number of WiFi access points or cellular base stations
$\mathcal{A}$	$N \times N$ association matrix that indicates D2D connections among devices over SR
a_{ij}	binary variable that indicates if device i has device j as cluster head
$\mathcal{B}$	$M \times N$ association matrix between APs and devices over LR
b_{mi}	binary variable that indicates if device i is associated to AP m
C_i	Cost function that models damage caused to network upon the failure of device i being assigned as cluster head
$P_{L,mi}$	Power received by device i from AP m over LR
$P_{s,ij}$	Power received by device j from device i over SR D2D
σ^2	Total noise level over any connection
Γ_i	Reliability metric of a given device i
ρ	Tradeoff between network reliability and performance
Δ_{LR}	Maximum number of devices connected to one AP
Δ_{SR}	Maximum number of devices connected to one cluster head
θ	Outage probability set by network operator
SNR_{LR}	minimum SNR threshold on LR
SNR_{SR}	minimum SNR threshold on SR

communications links. Thus, data can be received either over a LR link directly from the AP or SR link from a neighboring node in $\mathcal{C}$.

III. PROBLEM FORMULATION

We formulate the problem of reliable cluster formation with performance constraints as an integer linear programming problem. We define $N \times N$ association matrix $\mathcal{A}$ such that a_{ij} is a binary value that determines if device i is sharing data with device j over a SR connection. Moreover, we define an $M \times N$ association matrix $\mathcal{B}$ such that b_{mi} is a binary value that determines if AP m is transmitting data directly to device i over a LR connection. We denote by $P_{L,mi}$ the power received by device i from AP m over a LR connection and by $P_{s,ij}$ the power received by device j from neighboring device i over a SR D2D connection.

The goal is to select intelligently a set of devices $\mathcal{C}$ as cluster heads while jointly optimizing network reliability and performance. We optimize performance by maximizing the received power level at all devices irrespective if over LR connections from an AP or over SR connection from another nearby device. The received power level can then be mapped to data rate using $R = W \log_2(1 + \text{SNR})$ where W is bandwidth, $\text{SNR} = P_{\text{received}}/\sigma^2$, P_{received} is received power level either on SR or LR connection, and σ^2 represents total noise level. In addition to improving performance in terms of increased bit rate, the clustering approach also leads to energy savings while transmitting data among devices. As the received power level is maximized, devices within a cluster are selected in close proximity to each other and can transmit their data with reduced power leading to major energy reduction gains.

In order to model network reliability, we define a generic cost function C_i that models the damage caused to the network when device i acting as cluster head fails while sharing content with other nearby devices in its cluster. We model the failure cost of device i as a function of its individual reliability metric

Γ_i defined in (1) in addition to the number of devices it serves on the SR, i.e., number of devices in its cluster. Devices with relatively lower battery levels are allocated a reduced reliability metric and, thus, the solution opts to eliminate them from being selected as cluster heads. In addition, when the number of devices served by a cluster head increases, its failure cost grows as it impacts a larger pool of devices. The failure cost C_i can be expressed as follows:

$$C_i = (1 - \Gamma_i) \times \sum_{j=1, j \neq i}^N a_{ij} \quad (2)$$

The optimization problem can then be formulated as follows:

$$\text{minimize } \rho \sum_{i=1}^N C_i - \sum_{m=1}^M \sum_{i=1}^N b_{mi} P_{l,mi} - \sum_{i=1}^N \sum_{j=1, j \neq i}^N a_{ij} P_{s,ij} \quad (3)$$

subject to

$$a_{ij} \leq \sum_{m=1}^M b_{mi} \quad i, j = 1, \dots, N \quad (4)$$

$$\sum_{m=1}^M b_{mi} + \sum_{j=1, j \neq i}^N a_{ji} \leq 1 \quad i = 1, \dots, N \quad (5)$$

$$\sum_{i=1}^N b_{mi} \leq \Delta_{LR} \quad m = 1, \dots, M \quad (6)$$

$$\sum_{j=1}^N a_{ij} \leq \Delta_{SR} \quad i = 1, \dots, N \quad (7)$$

$$\sum_{m=1}^M \sum_{i=1}^N b_{mi} + \sum_{i=1}^N \sum_{j=1}^N a_{ij} \geq (1 - \theta)N \quad (8)$$

$$\frac{P_{l,mi}}{\sigma^2} \geq b_{mi} \text{SNR}_{LR} \quad \forall m, i \quad (9)$$

$$\frac{P_{s,ij}}{\sigma^2} \geq a_{ij} \text{SNR}_{SR} \quad \forall i, j \quad (10)$$

The objective function in (3) balances a trade-off between maximizing the overall received power levels over all LR and SR connections and minimizing the failure cost for each cluster head device in $\mathcal{C}$. The parameter ρ is introduced to balance the tradeoff between network reliability and performance; for instance, if ρ is large, then the solution would favor producing a more reliable network at the cost of overall performance and vice-versa.

The first constraint in (4) specifies that device i can transmit data to device j over a SR connection only when it is receiving its content from an AP over a LR connection. We limit the level of cooperation to two hops for practical feasibility reasons, especially in use cases where devices are low end IoT sensors. The second constraint in (5) limits the active reception data links of any device to one; it is satisfied if and only if a device is receiving data over either a LR link or SR link but not both simultaneously. Constraints represented in (6) and (7)

limit the number of devices connected to the AP and to a cluster head to Δ_{LR} and Δ_{SR} , respectively; these constraints are added to capture the limit on the number of transmission channels per AP or device. The constraint in (8) bounds the outage probability in the network; the number of served devices is set to a minimum of $(1 - \theta)N$ where θ represents the outage probability set by the network operator. Finally, the constraints in (9) and (10) require the received SNR per device to be above a minimum threshold on both LR (SNR_{LR}) and SR (SNR_{SR}) connections in order to maintain a target level of performance guarantee. The values of the various thresholds and parameters can be configured depending on the application scenario and requirements.

The formulated problem is an integer linear programming (ILP) problem and can be mapped to the K-medoids problem, which is $\mathcal{NP}$ -hard [33], [34]. The K-medoids problem entails selecting K centroids among devices such that the aggregated distance between the centroids and devices is minimized. Our problem can actually be reduced to K-medoids by relaxing problem constraints. For instance, in the case of a single AP, we set the reliability of all devices to 1 and set system parameters $\rho, \theta, \Delta_{LR}, \Delta_{SR}$ to 0, 0, K , and N , respectively. Consequently, we are only looking for centroids that minimize the distance between them and other nodes, which maps to the K-medoids problem. In addition, our problem becomes harder with multiple APs since centroid selection is also affected by the deployed position of the APs, as we attempt to also minimize the distance between each AP and the centroids (cluster heads) that it serves. To this end, we generate in the results section optimal solutions only for relatively small scale scenarios. In addition, we propose in the following section an efficient proactive heuristic algorithm for reliable cluster formation that is scalable to large scenarios with close-to-optimal performance.

IV. SOLUTION METHODOLOGY AND ALGORITHMS

The problem formulation in (3) can be solved for relatively small scale network scenarios to generate optimal reliable clustering results. For large scale network scenarios, we propose the utilization of a clustering algorithm based on the notion of electrostatic forces [35]. The authors in [35], [36] demonstrated the validity and effectiveness of an approach based on electrostatic forces for applications that include general data clustering and tumor detection in images. In our work, we capitalize on this approach and extend it to address the problem of reliable cluster formation in cooperative wireless networks.

The algorithm starts by randomly scattering K virtual centroids on the given area and then iteratively utilizes the law of electrostatics to determine optimized positions for the centroids; we describe these centroids as “virtual” since their locations need not overlap with existing devices in the network which requires an extension phase of the algorithm implementation to map each “virtual” centroid to one of the existing devices that would serve as a cluster head. In the algorithm, each device is assigned a negative fixed charge while virtual centroids are assigned dynamic positive charges;

hence, the force among centroids is repulsive, while the force between centroids and devices is attractive. In the context of our problem, we let the charge of each device be a function of its own reliability function as follows:

$$Q_i = -\Gamma_i. \quad (11)$$

This allows devices to be clustered based on their reliability and relative positions. As for identifying the best centroid locations, we set the charge of a given centroid k to be inversely proportional to the number of devices N_k associated to it. Thus, the charge of centroid k changes in every iteration as follows:

$$Q_k = \frac{\lambda}{N_k + 1}. \quad (12)$$

where λ is a pre-set parameter with value between 0 and 1; it can be configured to control the distance between the virtual centroid locations in the final solution.

In the network, there exist two main types of forces, namely, repulsion and attraction. Virtual centroids repel each other because they carry like charges while a centroid and a device attract each other because they carry unlike charges. Based on Coulomb's law, the force experienced by centroid k in the vicinity of centroid j can be calculated as follows using a unit vector to determine the force direction:

$$\vec{F}_{jk} = \kappa \frac{Q_k Q_j}{d_{k,j}^2} \times \frac{\vec{c}_k - \vec{c}_j}{\|\vec{c}_k - \vec{c}_j\|}, \quad (13)$$

where κ is Coulomb's constant, $d_{k,j}$ is the Euclidean distance between k and j , and $\vec{c}_k$ and $\vec{c}_j$ represent the centroid coordinate vectors, respectively.

The electrostatic force of attraction exerted by device j on centroid k is similarly calculated as follows:

$$\vec{F}_{jk} = \kappa \frac{Q_k Q_j}{d_{k,j}^2} \times \frac{\vec{c}_k - \vec{p}_j}{\|\vec{c}_k - \vec{p}_j\|}. \quad (14)$$

where $\vec{p}_j$ is the device coordinate vector.

Consequently, an electric field is formed among the nodes causing centroids to repel from each other and attract to devices. Hence, centroids move until electrostatic equilibrium is reached where the sum of forces is balanced and centroids are fixed. We denote the total force exerted on centroid k as $\vec{F}_k$ and calculate it as the summation of all repulsion and attraction forces as follows:

$$\begin{aligned} \vec{F}_k &= \sum_{j \neq k} \vec{F}_{jk} \\ &= \sum_{j \neq k, j \in \mathcal{C}} \kappa \frac{Q_k Q_j}{d_{k,j}^2} \times \frac{\vec{c}_k - \vec{c}_j}{\|\vec{c}_k - \vec{c}_j\|} + \sum_{j \in \mathcal{D}} \kappa \frac{Q_k Q_j}{d_{k,j}^2} \times \frac{\vec{c}_k - \vec{p}_j}{\|\vec{c}_k - \vec{p}_j\|} \end{aligned} \quad (15)$$

The direction in which centroids move is determined according to the sum of forces exerted on each centroid with a pre-configured step size η . Fixing the step size, we can calculate the centroid's new position as follows:

$$\vec{c}_k^{\tau+1} = \vec{c}_k^{\tau} + \eta \frac{\vec{F}_k}{\|\vec{F}_k\|}, \quad (16)$$

where τ represents the iteration number reflecting the algorithm's execution over time. The algorithm runs over multiple iterations till it reaches a stable state, whereby the virtual centroids positions vary only locally within a circle of small radius. Therefore, the algorithm's implementation keeps track of the position variation increments per centroid to decide when to stop and generate a solution.

The presented solution approach is denoted as RForce and is divided into three main phases: Phase 1, summarized in Algorithm 1 and Algorithm 2, optimizes the locations of virtual centroids using proposed approach based on electrostatic forces; Phase 2, summarized in Algorithm 3, maps the virtual centroids to existing devices that will act as cluster heads; and Phase 3, summarized in Algorithm 4, associates cluster heads with APs in an optimized way that enhances the download bit rate while satisfying the given constraints.

Algorithm 1 is called with two main inputs, a vector of randomly generated centroids and a vector of devices with given locations; the devices are the set of sensors or users deployed in the area of interest. The algorithm then goes into a loop that only halts when the centroids reach a stable state whereby their positions are nearly no longer varying. Inside the loop, the algorithm initially starts by resetting all associations between centroids and devices that have been done in the previous iteration by setting the centroids degree to zero and finding the best set of devices for each centroid. To do this, the algorithm utilizes ASSOCIATE_CENTROIDS method which loops through the devices and finds for each device the nearest centroid using the BEST_CENTROID method taking into account the constraint on degree bound. After finding the best centroid for each device, each centroid c_k gets its own vector of associated devices which form the set $\mathcal{A}_k$; hence, $\mathcal{A}_k$ with c_k form an initial cluster. After obtaining the corresponding centroid of each device, we can now calculate the forces exerted on each centroid by the set of devices within its cluster and by the other centroids, using (15). Then the centroid is moved according to (16).

In Phase 2 of the algorithm, the virtual centroids with optimized locations are mapped to existing devices that will act as cluster heads using the intelligence summarized in Algorithm 3. It starts by defining a set $\mathcal{M}$ that stores the indices of the cluster head node for each device d_i . Initially, d_i is set to -1 which means that this set does not belong to any group. We then start by looping on the set of centroids and finding for each virtual centroid a corresponding device that is not connected to a cluster and is within minimal distance to achieve high communications quality, using the BEST_DEVICE method. Finally, after mapping each virtual centroid to a cluster head device, we shift all the virtual centroid's associated devices to the corresponding cluster head device to form a cluster.

In Phase 3 of the algorithm, all cluster heads need to be connected to the available set of APs as our system model assumes an infrastructure based network scenario. The allocation should maximize performance quality while respecting the constraint on the number of LR connections per AP. One can address this using a standard MST (Minimum Spanning Tree) algorithm such as Prim or Kruskal in order to ensure

Algorithm 1 RFORCE - PHASE 1

```

1: procedure RFORCE( $\mathcal{C}$ ,  $\mathcal{D}$ )
2: INPUT: Set of centroids  $\mathcal{C}$  and set of devices  $\mathcal{D}$ 
3: OUTPUT: Association matrix  $\mathcal{A}$  and optimized location of
   all centroids
4:   while NOT_STABLE( $\mathcal{C}$ ) do
5:     for each centroid  $c_k \in \mathcal{C}$  do
6:        $c_k.\text{degree} \leftarrow 0$ 
7:     end for
8:      $\mathcal{A} \leftarrow \text{ASSOCIATE\_CENTROIDS}(\mathcal{C}, \mathcal{D})$ 
9:      $\mathcal{F} \leftarrow \emptyset$ 
10:    for each centroid  $c_k \in \mathcal{C}$  do
11:      for each node  $v \in \mathcal{C} \cup \mathcal{A}_k$  do
12:         $\mathcal{F}_k \leftarrow \mathcal{F}_k + \text{FORCE}(c_k, v)$ 
13:      end for
14:    end for
15:     $\text{MOVE\_CENTROIDS}(\mathcal{C}, \mathcal{F})$ 
16:  end while
17: end procedure
18:
19: procedure NOT_STABLE( $\mathcal{C}$ )
20: INPUT: Set of centroids  $\mathcal{C}$ 
21: OUTPUT: TRUE if at least one centroid  $c_k$  changed its
   position, FALSE otherwise
22:   for each centroid  $c_k \in \mathcal{C}$  do
23:     if  $c_k.x \neq c_k.x'$  OR  $c_k.y \neq c_k.y'$  then
24:       return TRUE
25:     end if
26:   end for
27:   return FALSE
28: end procedure
29:
30: procedure MOVE_CENTROID( $\mathcal{C}$ ,  $\mathcal{F}$ )
31: INPUT: Set of centroids  $\mathcal{C}$ , and corresponding force  $\mathcal{F}$ 
   exerted on them
32: OUTPUT: Updated coordinates of each centroid
33:   for each centroid  $c_k \in \mathcal{C}$  do
34:      $c_k.x' = c_k.x$ 
35:      $c_k.y' = c_k.y$ 
36:      $c_k.x' = c_k.x + \eta \frac{\mathcal{F}_k.x}{\|\mathcal{F}_k\|}$ 
37:      $c_k.y' = c_k.y + \eta \frac{\mathcal{F}_k.y}{\|\mathcal{F}_k\|}$ 
38:   end for
39: end procedure

```

that one cluster head receives its content from only one of the available APs. However, since the APs have a degree bound of Δ_{LR} , this problem reduces to MBDST (Minimum Bounded Degree Spanning Tree) which is known to be NP-hard [37]. Thus, we resort to a modified version of Kruskal's algorithm to associate cluster heads to APs without violating the degree constraint of the APs and the tree property (see Algorithm 4 for the details).

V. PERFORMANCE RESULTS AND ANALYSIS

In this section, we demonstrate the effectiveness of the proposed proactive and scalable approach for reliable cluster

Algorithm 2 RFORCE - PHASE 1 *cont'd*

```

ASSOCIATE_CENTROIDS AND BEST_CENTROID METHODS
40: procedure ASSOCIATE_CENTROIDS( $\mathcal{C}$ ,  $\mathcal{D}$ )
41: INPUT: Set of centroids  $\mathcal{C}$  and set of devices  $\mathcal{D}$ 
42: OUTPUT: Association matrix  $\mathcal{A}$  that indicates devices
   mapped to each centroid
43:    $\mathcal{A} \leftarrow \emptyset$ 
44:   for each device  $d \in \mathcal{D}$  do
45:      $c_k \leftarrow \text{BEST\_CENTROID}(\mathcal{C}, d)$ 
46:      $c_k.\text{degree} \leftarrow c_k.\text{degree} + 1$ ;
47:      $\mathcal{A}_k \leftarrow \mathcal{A}_k \cup d$ 
48:   end for
49:   return  $\mathcal{A}$ 
50: end procedure
51:
52: procedure BEST_CENTROID( $\mathcal{C}$ ,  $d$ )
53: INPUT: Set of centroids  $\mathcal{C}$  and certain device  $d$ 
54: OUTPUT: Best centroid for device  $d$ 
55:    $\text{best\_centroid} \leftarrow \emptyset$ 
56:    $\text{min\_distance} \leftarrow \infty$ 
57:   for each centroid  $c_k \in \mathcal{C}$  do
58:      $\text{distance} \leftarrow \text{DISTANCE}(c_k.x, c_k.y, d.x, d.y)$ 
59:     if  $c_k.\text{degree} < \Delta_{\text{SR}}$  AND  $\text{distance} < \text{min\_distance}$ 
   then
60:        $\text{min\_distance} \leftarrow \text{distance}$ 
61:        $\text{best\_centroid} \leftarrow c_k$ 
62:     end if
63:   end for
64:   return  $\text{best\_centroid}$ 
65: end procedure

```

formation in cooperative wireless networks using a combination of optimization problem solutions for relatively small scale network scenarios, Monte-Carlo simulation results based on the presented low complexity algorithms in Section IV, and experimental test bed results to demonstrate feasibility under realistic operational conditions. For the optimization problem solution, we use the `intlinprog` mixed integer linear programming function in Matlab (denoted as `ROptimal` in the sequel). We also compare the RForce algorithm to the standard kMeans clustering algorithm in terms of reliability, download bit rates, and execution complexity. For kMeans clustering, we use Lloyd's algorithm [38] with slight modification to capture our problem constraints. We consider several network scenarios and vary system parameters to produce a wide range of results that allow for extracting insights and capturing tradeoffs. For the Monte-Carlo simulation results, we average over 25 runs for each set of network scenario and system parameters and plot average performance metrics. Table II summarizes the values used in the results for key network and algorithm parameters.

Fig. 2 compares the running time in msec on a standard PC for the different approaches assuming a network scenario with one AP and number of devices ranging from 50 to 250. This clearly demonstrates the gap in complexity between generating the optimal solution and generating solutions using our proposed RForce algorithm and the standard kMeans

Algorithm 3 RFORCE - PHASE 2

```

1: procedure MAP_CENTROIDS( $\mathcal{C}, \mathcal{D}, \mathcal{A}$ )
2: INPUT: Set of centroids  $\mathcal{C}$ , set of devices  $\mathcal{D}$ , and associa-
   tion matrix  $\mathcal{A}$ 
3: OUTPUT: Cluster head mapping vector  $\mathcal{M}$  that denotes
   the index of the device that acts as cluster head of each
   device
4:    $\mathcal{M} \leftarrow \emptyset$ 
5:   for each device  $d_i \in \mathcal{D}$  do
6:      $\mathcal{M}_i \leftarrow -1$ 
7:   end for
8:   for each centroid  $c_k \in \mathcal{C}$  do
9:      $d_j \leftarrow \text{BEST\_DEVICE}(\mathcal{D}, \mathcal{M}, c_k)$ 
10:    ASSOCIATE_DEVICES( $\mathcal{A}, \mathcal{M}, d_j$ )
11:   end for
12: end procedure
13:
14: procedure BEST_DEVICE( $\mathcal{D}, \mathcal{M}, c$ )
15: INPUT: Set of devices  $\mathcal{D}$ , cluster head mapping vector  $\mathcal{M}$ ,
   and specific centroid  $c$ 
16: OUTPUT: Best device that can play the role of the input
   virtual centroid  $c$ 
17:   best_device  $\leftarrow \emptyset$ 
18:   min_distance  $\leftarrow \infty$ 
19:   for each device  $d_i \in \mathcal{D}$  do
20:     distance  $\leftarrow \text{DISTANCE}(c.x, c.y, d_i.x, d_i.y)$ 
21:     if  $\mathcal{M}_i == -1$  AND distance  $<$  min_distance then
22:       min_distance  $\leftarrow$  distance
23:       best_device  $\leftarrow d_i$ 
24:     end if
25:   end for
26:   return best_device
27: end procedure
28:
29: procedure ASSOCIATE_DEVICES( $\mathcal{A}, \mathcal{M}, d_j$ )
30: INPUT: Association matrix  $\mathcal{A}$ , cluster head mapping vector
    $\mathcal{M}$ , and device  $d_j$ 
31: OUTPUT: Updated cluster head mapping vector  $\mathcal{M}$  to
   indicate  $d_j$  as the cluster head of all devices assigned to
   the corresponding virtual centroid, where  $d_j$  is output by
   BEST_DEVICE
32:    $\mathcal{M}_j \leftarrow j$ 
33:   for each device  $d_i \in \mathcal{A}_j$  do
34:      $\mathcal{M}_i \leftarrow j$ 
35:   end for
36: end procedure
37:

```

algorithm; the computational complexity of the optimal solution increases exponentially with feasible outcome only for network scenarios having up to 100 devices. On the other hand, RForce and kMeans are shown to have similar low complexity and fast execution time which makes both of them applicable to ultra dense network scenarios with dynamic adaptation over time.

Fig. 3 and Fig. 4 compare the performance of the different algorithms in terms of communications quality and reliability

Algorithm 4 RFORCE - PHASE 3

```

1: procedure RFORCE - MST( $\mathcal{S}, \mathcal{H}$ )
2: INPUT: Set of access points  $\mathcal{S}$ , and set of cluster heads  $\mathcal{H}$ 
3: OUTPUT: Association matrix  $\mathcal{B}$  that maps each cluster
   head to one access point
4:    $Q \leftarrow \emptyset$ 
5:    $\mathcal{B} \leftarrow \emptyset$ 
6:   for each  $s \in \mathcal{S}$  do
7:     for each  $h \in \mathcal{H}$  do
8:        $Q.\text{push}(s, h)$ 
9:     end for
10:   end for
11:   while  $Q$  is not empty do
12:      $s, h = Q.\text{pop}()$ 
13:     if  $s.\text{degree} < \Delta_{\text{LR}}$  then
14:        $\mathcal{B}_h \leftarrow s$ 
15:        $s.\text{degree} \leftarrow s.\text{degree} + 1$ 
16:     end if
17:   end while
18:   return  $\mathcal{B}$ 
19: end procedure
20:

```

TABLE II: Network and algorithm parameters

Parameter	Value
Area size	100m by 100m
AP transmit power	10 Watts
Device transmit power	0.22 Watts
β	0.3
σ^2	1×10^{-9} Watts
ρ	20
θ	0.05
Δ_{LR}	30 connections per AP
Δ_{SR}	10 connections per cluster head
SNR_{LR}	1
SNR_{SR}	1
λ	0.8
η	0.4
$\gamma_i \forall i$	1

for network scenarios with one AP and four APs, respectively, assuming the number of devices to be served is between 50 and 250. The results demonstrate the effectiveness of the proposed RForce approach in achieving network reliability close to the optimal solution and much higher than the kMeans approach; this is reflected in the plots showing the failure cost metric, defined in (2), as a function of the number of devices. The results also demonstrate that this effectiveness in reliability is not negatively impacting communications bit rate or download speed, as the average bit rate on the SR links is shown to be close between RForce and kMeans, with both worse than the optimal reliable clustering solution. For example, assuming 200 devices in a given area with one AP, clusters formed by kMeans lead to around triple the failure cost compared to the solution produced by RForce, at the expense of around 0.5 Mbps reduction in average SR bit rate (around 10% reduction only); this is expected since kMeans aims at maximizing network throughput without accounting for reliability. When compared to the optimal solution, RForce

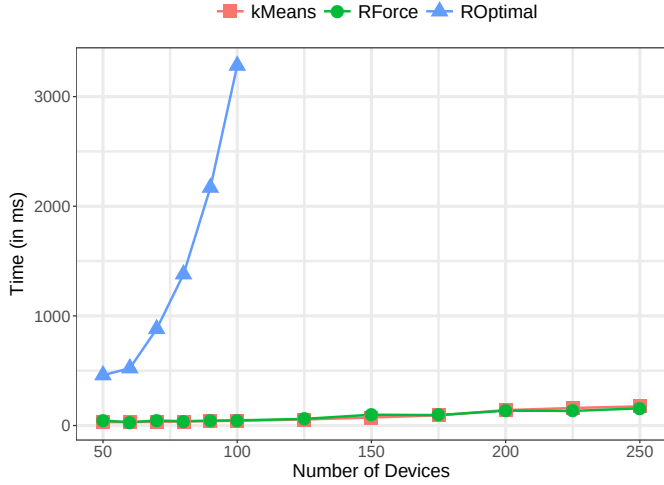


Fig. 2: Average execution time of various approaches assuming a network scenario with one AP.

is shown to have relatively close performance for both bit rate and reliability, yet with much higher execution speed as shown in Figure 2.

The same trends and insights on the effectiveness of RForce are also demonstrated in Fig. 4 with four APs; compared to the results with one AP, the bit rates become higher since more devices are served by the four APs on LR connections with closer proximity, whereas the failure cost also increases since more devices are chosen as cluster heads due to their proximity to the APs and some of these chosen cluster heads do not have high reliability metrics. These observations are consistent with the tradeoff between bit rate and reliability as captured in the objective function of the optimization problem formulation in (3).

We have also compared the various algorithms in terms of average network LR bit rate with results shown in Fig. 5 assuming network scenarios with one and four APs having between 50 and 250 devices. Similar to the SR bit rates, the results produced by RForce are almost equivalent to kMeans and both are not far from ROptimal. These results demonstrate the effectiveness of Algorithm 4 in associating cluster heads with APs with high bit rates on the LR connections.

Next, we present sample snapshot network scenarios with results generated using RForce to provide additional insights on number of clusters, their locations, and selection of cluster heads. Fig. 6a presents one sample network scenario with 200 devices uniformly distributed in a $100m \times 100m$ area. We represent APs as red squares, cluster heads as orange triangles, connected devices as blue circles, and unconnected devices in outage as gray circles. We also designate the reliability factor of each device by its opacity level. The figure shows that highly reliable devices (darker colors) are selected as cluster heads and they are also well positioned among the devices they serve in their cluster. Fig. 6b shows another example scenario with four APs and 200 devices. We notice that RForce tends to select more cluster heads when the number of APs increases as the algorithm tends to choose more devices as cluster heads to be served directly by the APs due to their proximity. It is

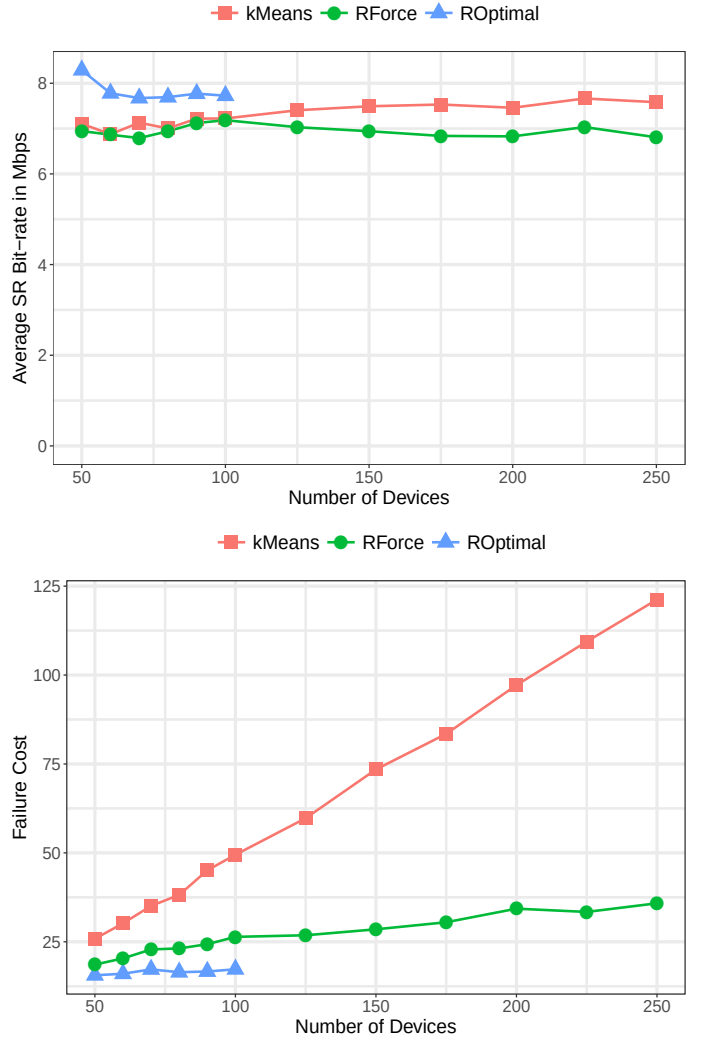


Fig. 3: Top: Average network SR bit rate in Mbps versus number of devices assuming one AP; Bottom: Average failure cost versus number of devices assuming one AP.

shown also that there are clusters composed of a single device served by in its own by the AP in order to avoid an increase in outage rate. Fig. 7 compares the resulting clusters between RForce and kMeans approaches; these plots show clearly the relative limitations of kMeans as devices with low reliability are chosen as cluster heads and outage rate is higher with more devices not served.

A. Energy Efficiency and Scalability

In order to provide a more tangible quantification of the impact of the failure cost metric, we estimate the lifetime of each selected cluster head based on a given set of assumptions taking into account available battery budget. We consider devices powered with batteries of 2000 mAh capacity and use the battery consumption model presented in [39]; to this end, we assume that devices are downloading data continuously with power consumption 1.27 W . We then derive the current in Ampere to be 0.34 A assuming a nominal voltage of 3.7 V . Battery lifetime is then calculated as $E * 2000\text{ mAh} / 0.34\text{ A}$,

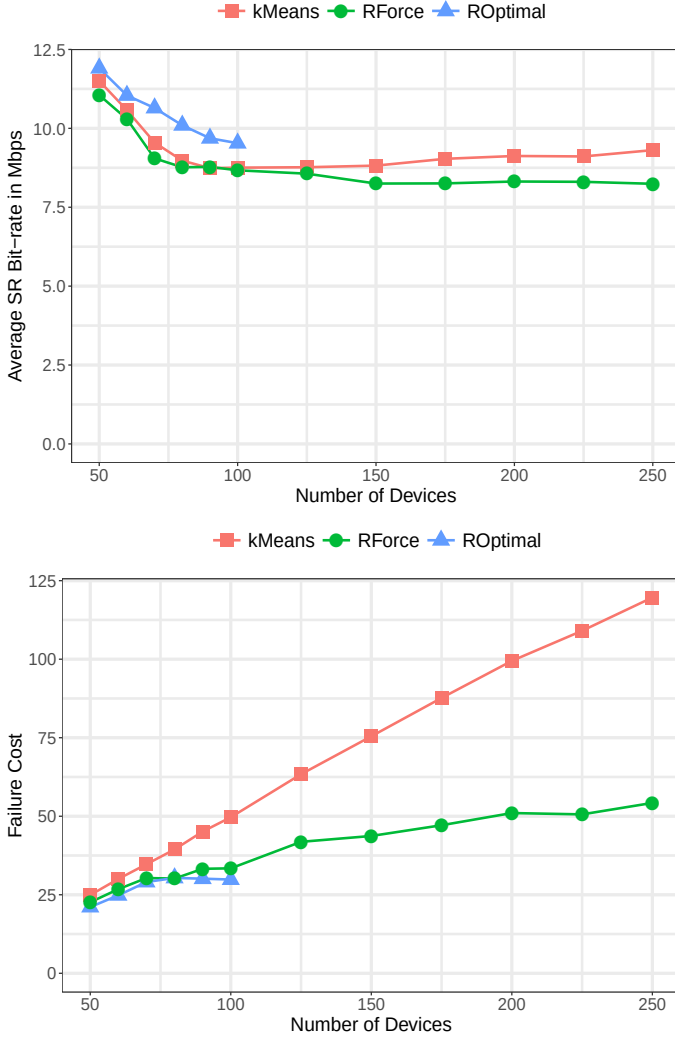


Fig. 4: Top: Average network SR bit rate in Mbps versus number of devices assuming four APs; Bottom: Average failure cost versus number of devices assuming four APs.

where E is the actual battery indicator in percentage. In our simulations, we assigned to the devices arbitrary values of E ranging from 10% to 90% to mimic a realistic network scenario.

Fig. 8 plots the resulting average lifetime of all cluster heads selected by each algorithm. RForce outperforms kMeans as the number of devices varies with close performance to ROptimal. We note that the average cluster head lifetime of both RForce and ROptimal increase as the number of devices increases within the given area due to the fact that they tend to select the most reliable cluster heads and more of these become available as the number of devices increases in the given area of interest. The performance of kMeans, however, is almost constant with limited variability as the number of devices increases since the algorithm does not account for reliability during the clustering process.

Finally, Fig. 9 presents results for ultra dense network scenario with up to 4000 devices in order to demonstrate the scalability of the proposed RForce approach and reconfirm its

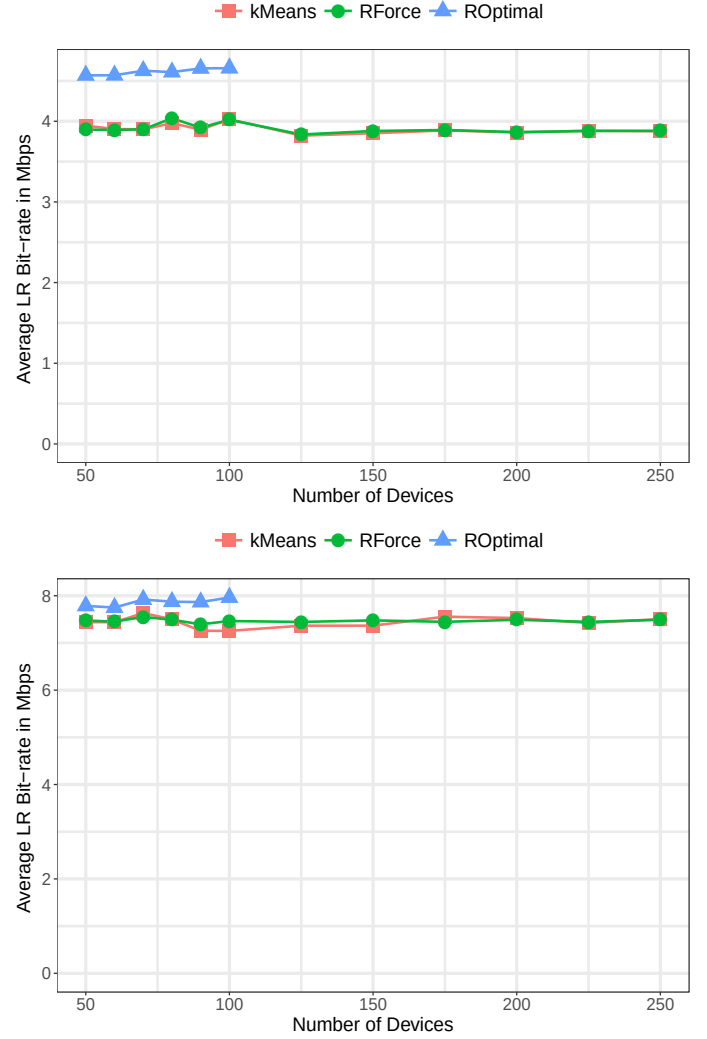


Fig. 5: Top: Average network LR bit rate in Mbps versus number of devices assuming one AP; Bottom: Average network LR bit rate in Mbps versus number of devices assuming four APs.

superiority in terms of reliability compared to the standard kMeans approach. It is important to note that it was not feasible to generate optimal results for such high density network scenarios due to high complexity.

B. Experimental Evaluation using Test Bed Measurements

In this section, we demonstrate the feasibility of implementing the RForce algorithm in a real experimental test bed and evaluate its operation under realistic network conditions. Our test bed consists of a set of devices that are connected via a WiFi access point to download content from a remote server. The devices run software applications that facilitate formation of clusters with coordination from a centralized management server. In our implementation, we used Android smartphones to mimic the devices as a proof of concept since smartphones can be programmed to communicate over two interfaces and serve as relays in cooperative networks, e.g., using WiFi or cellular over LR with APs and WiFi-Direct or Bluetooth over SR directly with neighboring devices. Similar implementation

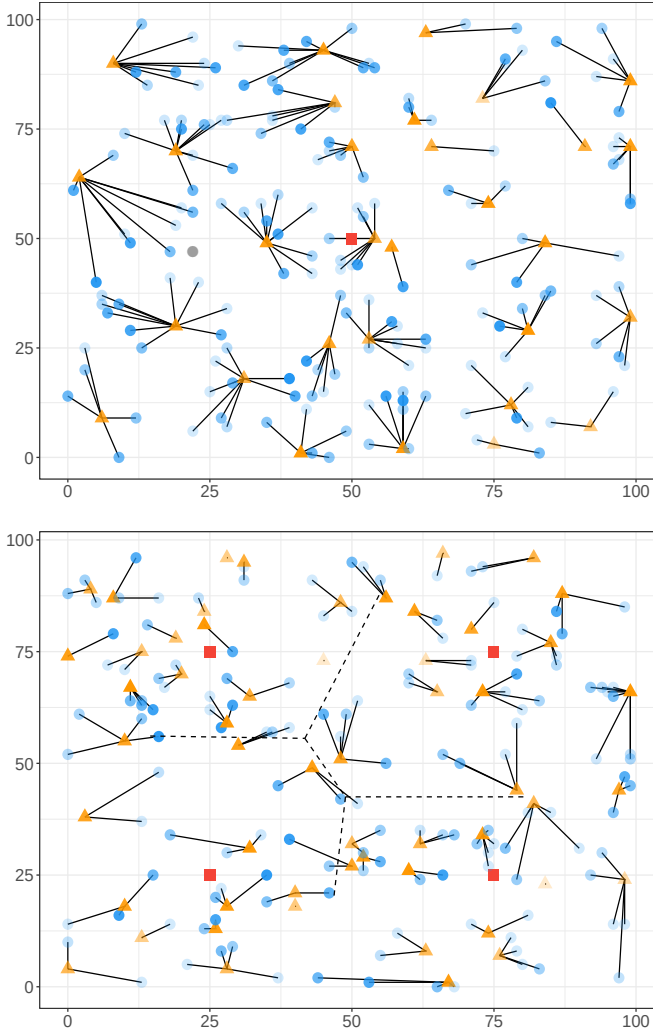


Fig. 6: Sample snapshot network scenarios generated by RForce assuming one AP (upper plot) and four APs (lower plot) with 200 devices.

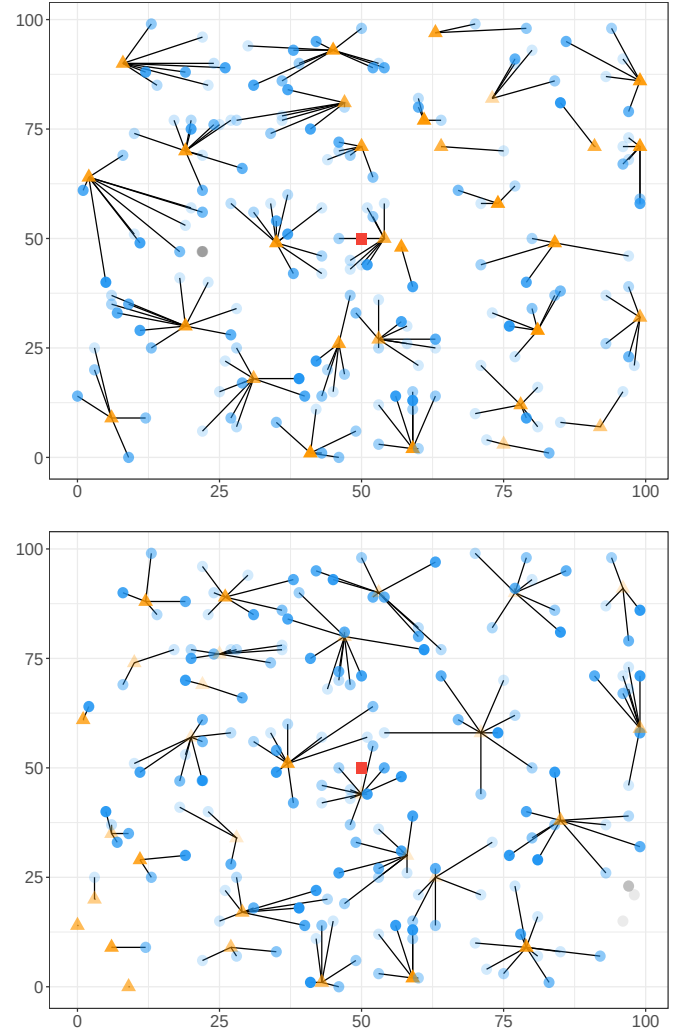


Fig. 7: Sample snapshot network scenarios assuming one AP and 200 devices with clustering using RForce (upper plot) and kMeans (lower plot).

can also be done on other types of devices such as sensors or IoT nodes.

The software running at the device level starts with a discovery phase where all devices scan the area and report to the management server various information including their IDs (e.g., MAC addresses), current battery capacity, list of discovered neighboring devices, and the captured received signal strength level from each neighboring device and AP. Fig. 10 presents a summary of the messages exchanged between a set of devices and the management server.

The server uses the received information from the existing devices to build an initial graph of the network as shown in Fig. 11 from an example real experiment with 12 devices and one AP. This graph is fully meshed because testing is done in a lab where all devices can see each other; it also shows the ID and battery indicator level as reported by each device in addition to a measure of the channel quality between all pairs of devices (the smaller the value on the lines connecting devices, the better the channel quality). After discovering all

devices, the server runs the proposed RForce algorithm on the graph to form reliable clusters of devices. The resulting outcome is shown in Fig. 12 with the devices divided into three clusters with cluster heads having relatively high battery capacity and, thus, high reliability. Note that in the middle cluster, the cluster head is not the best among the three nodes, but it was selected due to its better channel quality with respect to the AP and to the other devices; this demonstrates the tradeoff characteristics of our approach balancing reliability with communications bit rate.

To demonstrate how results vary between different network scenarios, we present in Fig. 13 and Fig. 14 the initial network graph and the resulting reliable clusters after running the RForce algorithm, respectively, using another testbed experiment with also 12 devices and one AP.

VI. CONCLUSIONS

In this work, we have proposed a proactive and scalable approach for reliable clustering in wireless networks with device-

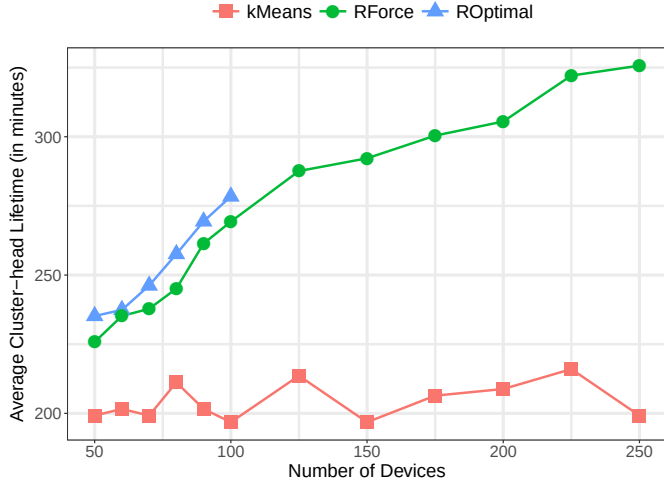


Fig. 8: Average cluster head lifetime in minutes as a function of the number of devices assuming a network scenario with one AP.

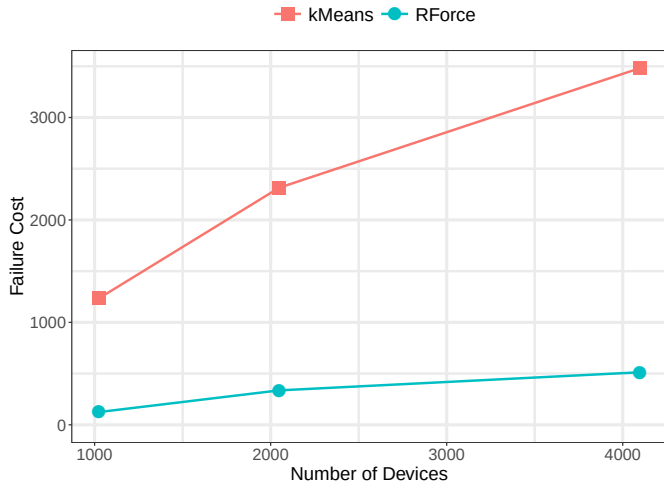


Fig. 9: Average failure cost for ultra dense network scenarios with up to 4000 devices.

to-device (D2D) offloading. Enhancing reliability in cooperative wireless networks is essential for practical implementation as devices (mobile users in a WiFi or cellular network, cars in a vehicular ad hoc network, or sensors in an IoT network) are not controlled by the network operator and, thus, can be dynamic in their mobility, computing and energy resources, and willingness to cooperate. Our proposed approach is characterized as flexible by capturing reliability metrics as part of the clustering process, generic by its application to different wireless network scenarios, and scalable by having low complexity and facilitating dynamic real time implementation. The solution methodology combines integer linear optimization problem formulation with heuristic algorithm design based on the notion of electrostatic forces. We utilize both extensive Monte-Carlo simulations and experimental test bed demonstrations in order to evaluate performance gains with respect to standard techniques and highlight interesting reliability-rate tradeoffs.

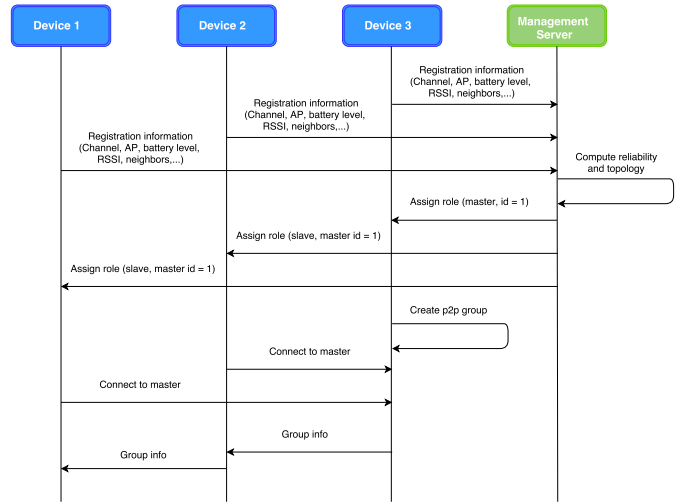


Fig. 10: Flowchart showing summary of message exchanges between devices and the management server in our testbed implementation.

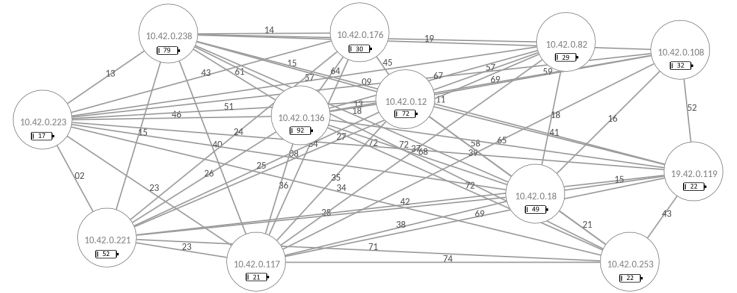


Fig. 11: Snapshot from the test bed server showing the initial graph with a total of 12 devices that need to stream content from a remote server.

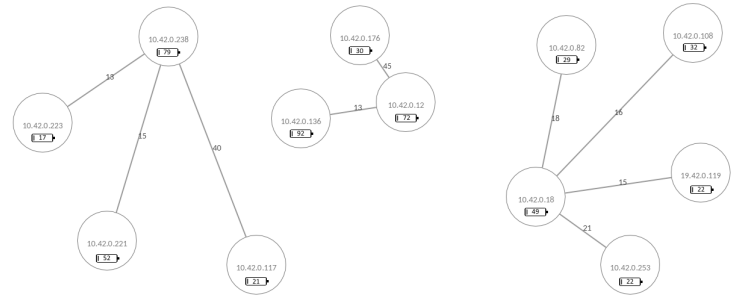


Fig. 12: Snapshot from the test bed server showing the clusters formed after executing the RForce algorithm.

The results confirm that it is practically feasible to enhance D2D cooperation reliability notably with a limited tradeoff cost in communication bit rate using the proposed efficient reliable clustering approach.

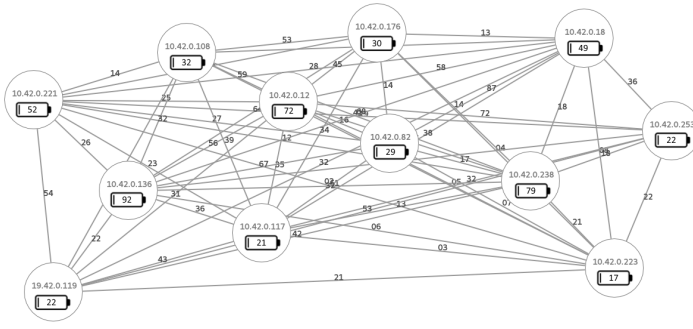


Fig. 13: Snapshot from the test bed server showing the initial graph with a total of 12 devices that need to stream content from a remote server.

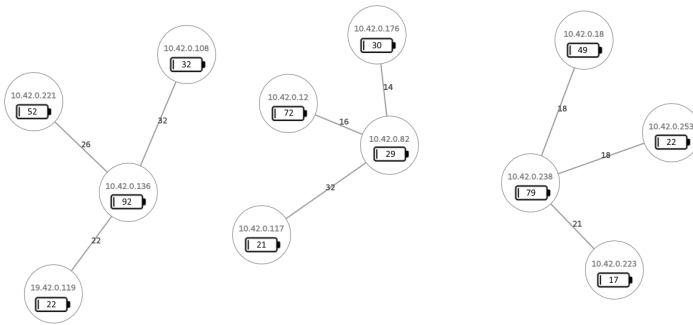


Fig. 14: Snapshot from the test bed server showing the clusters formed after executing the RForce algorithm.

ACKNOWLEDGEMENTS

This work is supported by the Lebanese American University under LAU fund # SRDC-r-2017-4.

REFERENCES

- [1] O. Bello and S. Zeasally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, pp. 1172–1182, September 2016.
- [2] P. Gandotra and R.K. Jha, "Device-to-device communication in cellular networks: A survey," *Elsevier Journal of Network and Computer Applications*, vol. 71, pp. 99–117, August 2016.
- [3] M.N. Tehrani and M. Uysal and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, pp. 86–92, May 2014.
- [4] D. Camps-Mur, A. Garcia-Saavedra and P. Serrano, "Device-to-device communications with Wi-Fi Direct: Overview and experimentation," *IEEE Wireless Communications*, vol. 20, pp. 96–104, June 2013.
- [5] G. Fodor and E. Dahlman and G. Mildh and S. Parkvall and N. Reidner and G. Miklos and Z. Turanyi, "Design aspects of network assisted device-to-device communications," *IEEE Communications Magazine*, vol. 50, pp. 170–177, March 2012.
- [6] A. Detti, B. Ricci and N. Blefari-Melazzi, "Mobile peer-to-peer video streaming over information-centric networks," *Elsevier Computer Networks*, vol. 81, pp. 272–288, April 2015.
- [7] K.-W. Lim, W.-S. Jung, and Y.-B. Ko, "Energy efficient quality-of-service for wlan-based d2d communications," *Ad Hoc Networks*, vol. 25, pp. 102–116, February 2015.
- [8] S. Mumtaza, H. Lundqvist, K. M. Saidul Huqa, J. Rodriguez and A. Radwan, "Smart Direct-LTE communication: An energy saving perspective," *Elsevier Ad Hoc Networks*, vol. 13, pp. 296–311, February 2014.
- [9] K. Jahed, M. Younes and S. Sharafeddine, "Energy measurements for mobile cooperative video streaming," *IFIP Wireless Days*, November 2012.
- [10] V. Sucasas, A. Radwan, H. Marques, J. Rodriguez, S. Vahid, and R. Tafazolli, "A survey on clustering techniques for cooperative wireless networks," *Ad Hoc Networks*, vol. 47, pp. 53–81, September 2016.
- [11] M. A. Khan, W. Cherif, F. Filali, and R. Hamila, "Wi-fi direct research - current status and future perspectives," *Journal of Network and Computer Applications*, vol. 93, pp. 245–258, September 2017.
- [12] N. Haider, M. Imran, M. Younis, N. Saad and M. Guizani, "A novel mechanism for restoring actor connected coverage in wireless sensor and actor networks," *IEEE ICC 2015 - Ad hoc and Sensor Networking Symposium*, June 2015.
- [13] M. Younis, I. F. Senturkb, K. Akkayab, S. Leec and F. Seneld, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Elsevier Computer Networks*, vol. 58, pp. 254–283, January 2014.
- [14] O. M. Al-Kofahi and A. E. Kamal, "Survivability strategies in multihop wireless networks," *IEEE Wireless Communications*, vol. 17, pp. 71–80, October 2010.
- [15] M. Garrosi, B. Zafar, and M. Haardt, "Prolonged network life-time in self-organizing peer-to-peer networks with e-rssi clustering," in *IEEE International Conference on Communications (IEEE ICC)*, June 2012.
- [16] A. Shahin and M. Younis, "Efficient multi-group formation and communication protocol for wi-fi direct," in *Conference on Local Computer Networks (LCN)*, October 2015.
- [17] M. Azharuddin, P. Kuila, and P. Jana, "Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks," *Computers & Electrical Engineering*, vol. 41, pp. 177–190, January 2015.
- [18] S. Sharafeddine, K. Jahed, O. Farhat, and Z. Dawy, "Failure recovery in wireless content distribution networks with device-to-device cooperation," *Computer Networks*, April 2017.
- [19] P. Chaki, M. Yasuda, and N. Fujita, "Seamless group reformation in wifi peer to peer network using dormant backend links," in *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, January 2015.
- [20] S. Pathak and S. Jain, "An optimized stable clustering algorithm for mobile ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 51, pp. 1–11, March 2017.
- [21] N. Mansoor, A. M. Islam, M. Zareei, S. Baharun, and S. Komaki, "Construction of a robust clustering algorithm for cognitive radio ad-hoc network," in *International Conference on Cognitive Radio Oriented Wireless Networks (CrownCom 2015)*, April 2015.
- [22] T. Shiva Prakash, K. Raja, K. Venugopal, S. Iyengar, and L. Patnaik, "Fault tolerant qos adaptive clustering for wireless sensor networks," in *Ninth International Conference on Wireless Communication and Sensor Networks*, 2014.
- [23] D. Jia, H. Zu, S. Zou, and P. Hu, "Dynamic cluster head selection method for wireless sensor network," *IEEE Sensors Journal*, vol. 16, pp. 2746–2754, April 2016.
- [24] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for wsn to extend the network lifetime," *IEEE Sensors Journal*, vol. 16, pp. 137–144, January 2016.
- [25] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.
- [26] W. Cherif, M. Khan, F. Filali, S. Sharafeddine, and Z. Dawy, "P2p group formation enhancement for opportunistic networks with wi-fi direct," in *IEEE Wireless Communications and Networking Conference (IEEE WCNC)*, March 2017.
- [27] K. Jahed, O. Farhat, G. Al-Jurdi, and S. Sharafeddine, "Optimized group owner selection in wifi direct networks," in *24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, September 2016.
- [28] U. Menegato, L. Cimino, S. Delabrida, F. Medeiros, R. Oliveira, and D. Ufop, "Dynamic clustering in wifi direct technology," in *12th ACM International Symposium on Mobility Management and Wireless Access (MobiWac 2014)*, September 2014.
- [29] K. Liu, W. Shen, B. Yin, X. Cao, L. Cai, and Y. Cheng, "Development of mobile ad-hoc networks over wi-fi direct with off-the-shelf android phones," in *IEEE International Conference on Communications (ICC)*, May 2016.
- [30] K. Jahed and S. Sharafeddine and A. Moussawi and A. Abou Daya and H. Dbouk and S. Kassir and Z. Dawy and P. Valsalan and W. Cherif and F. Filali, "Scalable multimedia streaming in wireless networks

- with device-to-device cooperation,” *ACM Multimedia 2016 Conference*, October 2016.
- [31] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: a measurement study and implications for network applications,” in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pp. 280–293, ACM, 2009.
 - [32] N. Abbas, Z. Dawy, H. Hajj, and S. Sharafeddine, “Energy-throughput tradeoffs in cellular/wifi heterogeneous networks with traffic splitting,” in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2294–2299, April 2014.
 - [33] C. H. Papadimitriou, “Worst-case and probabilistic analysis of a geometric location problem,” *SIAM Journal on Computing*, vol. 10, no. 3, pp. 542–557, 1981.
 - [34] N. Megiddo and K. Supowit, “On the complexity of some common geometric location problems,” *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.
 - [35] M. Khandani, P. Saeedi, Y. Fallah, and M. Khandani, “A novel data clustering algorithm based on electrostatic field concepts,” in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, March 2009.
 - [36] M. Khandani, R. Bajcsy, and Y. Fallah, “Automated segmentation of brain tumors in mri using force data clustering algorithm,” in *International Symposium on Visual Computing*, November 2009.
 - [37] M. X. Goemans, “Minimum bounded degree spanning trees,” in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06)*, October 2006.
 - [38] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, September 2006.
 - [39] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, “Demystifying 802.11n power consumption,” in *International Conference on Power Aware Computing and Systems*, 2010.