ESPP: Efficient Sector-based Charging Scheduling and Path Planning for WRSNs with Hexagonal Topology

Abdulbary Naji, Ammar Hawbani*, Xingfu Wang*, Haithm M. Al-Gunid, Yunes Al-Dhabi, Ahmed Al-Dubai, Amir Hussain, Liang Zhao,and Saeed Hamood Alsamhi

Abstract—Wireless Power Transfer (WPT) is a promising technology that can potentially mitigate the energy provisioning problem for sensor networks. In order to efficiently replenish energy for these battery-powered devices, designing appropriate scheduling and charging path planning algorithms is essential and challenging. Whilst previous studies have tackled this challenge, the conjoint influences of network topology, charging path planning, and energy threshold distribution in Wireless Rechargeable Sensor Networks (WRSNs) are still in their infancy. We mitigate the aforementioned problem by proposing novel algorithmic solutions to efficient sector-based on-demand charging scheduling and path planning. Specifically, we first propose a hexagonal cluster-based deployment of nodes such that finding an NP-Complete Hamiltonian path is feasible. Second, each cluster is divided into multiple sectors and a charging path planning algorithm is implemented to yield a Hamiltonian path, aimed at improving the Mobile Charging Vehicle (MCV) efficiency and charging throughput. Third, we propose an efficient algorithm to calculate the *importance* of nodes to be used for charging duration decision-making and prioritization. Fourth, a non-preemptive dynamic priority scheduling algorithm is proposed for charging tasks' assignments and scheduling. Finally, extensive simulations have been conducted, revealing the significant advantages of our proposed algorithms in terms of energy efficiency, response time, dead nodes' density, and queuing processing.

Index Terms—wireless rechargeable sensor networks, wireless sensor networks, hexagonal-clustering, scheduling, wireless power transfer, path planning;

1 Introduction

In the rapid growth of the Internet of Things (IoT), applications from personal electronics to industrial machines and sensors connect wirelessly to the internet, covering a wide variety of use cases in various environments and serving diverse requirements. Wireless Sensor Networks (WSNs) [18] in conjunction with IoT enables distributed measurements across vast physical systems to effectively analyze everything from rain forests and river deltas to health and safety and from smart home to smart city. Usually, the sensory measurements are reported to the cloud through IoT-enabled gateways for further data analytics. One of the most important applications of WSN is in the Smart Grid, which is increasingly finding new ways to measure and communicate the flow of electricity throughout the grid to enable efficient, reliable service. The movement towards intelligent power promises does not only improve delivery, but it also offers cost-effeteness through improved infrastructure maintenance and a better understanding of consumption patterns.

Nodes [14] are energy-constrained and usually deployed in areas where battery replacement or recharging is difficult and risky. It is even more impractical and impossible to provide a power lines to power these devices. Many solutions have been proposed including energy harvesting [6, 16] collectors such as large solar panels and wind generators. But energy harvesting still infeasible solution since it does not provide sufficient energy in most environments. Moreover, the size and cost of the harvester impose a larger node size and high cost.

Recently, Wireless Power Transfer (WPT) [15, 19] has gained so much popularity after the experimental realization by Kurs et al. [7] it has been proved experimentally that a total of 60W electrical power can be transferred between two magnetically coupled coils over an air gap of 2m with 40% efficiency. WPT is a promising solution for energy provisioning [17] since the electrical energy can be transmitted from a power source to an electrical load across an air gap using induction coils. WPT provides more convenience, robustness, and greater accessibility to harsh environments. UAVs carrying WPT transmitter can easily log into a variety of risky places in risky situations.

Employing WPT in real-world applications still requires more research investigation not only deploying MCV/UAV carrying high-capacity battery but also charging requests handling, task scheduling algorithms, path planning algo-

Abdulbary Naji, A. Hawbani, X. Wang , Haithm M, and Yunes Al-Dhabi. are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: abdulbary@mail.ustc.edu.cn, wangxfu@ustc.edu.cn, anmande@ustc.edu.cn, haithmalgunid@mail.ustc.edu.cn,yunes@mail.ustc.edu.cn).
Liang Zhao is with the School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China (e-mail: lzhao@sau.edu.cn).
Ahmed Al-Dubai and Amir Hussain are with School of Computing, Edinburgh Napier University, United Kingdom E-mail: a.al-dubai@napier.ac.uk and A.Hussain@napier.ac.uk;
Saeed Hamood Alsamhi is with Insight Centre for Data Analytics, University of Galway, Galway, Ireland and IBB University, Ibb, Yemen E-mail:saeed.alsamhi@insight-centre.org
A. Hawbani and X. Wang are the corresponding authors

rithms, and queuing analysis. For most existing research in WRSN, researchers have mainly focused on the offline charging scheduling schemes in which the charging of individual nodes is carried out in a periodic and deterministic manner. Due to the uncertainty in both the energy demand and supply indicates, existing periodic charging solutions may suffer from non-negligible performance degradation [3].

For optimal efficiency in wireless power delivery, a feasible scheduling algorithm and optimal path planning algorithm are required. The charging tasks scheduler aims to achieve objectives such as maximizing charging throughput and minimizing the waiting time, latency, and response time. In practice, these objectives often conflict (e.g. energy efficiency versus response time). Thus a scheduler must implement an appropriate compromise depending on the system's objectives. Finding a schedulable algorithm that can guarantee that all nodes are charged before their deadline is infeasible or even NP-hard due to the velocity constraints of the mobile charge vehicles. Moreover, scheduling without path planning has the potential of wasting network resources. Due to the MCVs speed constraints, minimizing the traveling cost is an essential objective. Since the traveling-cost is directly proportional to the length of the charging path, the shortest Hamiltonian path is definitely regarded as an efficient and optimal traveling solution. A charging task is said to be schedulable if only if can be charged before its deadline. In other words, the mobile charger should respond faster to avoid the nodes' deadlines.

A better understanding of the network topology also plays a significant role in scheduling and path planning. The objective is not only finding the shortest Hamiltonian path but also considering the *importance* of the nodes (i.e., the nodes with higher degree centrality). For hexagonal-clustered-based topology, the *importance* of the nodes follows the *negative exponential distribution*, in which *importance* of the nodes diminishes as we move to higher layers within the cluster.

Motivations and Contributions: Despite previous studies have made significant efforts to tackle the aforementioned constraints, the influences of network topology, charging path planning, and energy threshold distribution on scheduling are not conjointly addressed. This motivates us to re-investigate and take a deep insight starting from the underlying network topology perspective to path planning and scheduling. For addressing these problems, we propose Efficient Sectored-based Charging Scheduling and Path Planning (ESPP), including the following contributions:

1) We propose a hexagonal-clustered-based network topology for WRSN, and develop a mathematical model for network deployment and planning. Moreover, our study proposes a cluster construction algorithm for nodes' deployment, and a Hexagonal-based low-cost connection algorithm for topology construction aiming at providing a strongly connected network and improving network performance and resource allocation.
2) To gain control over charge request arrivals and provide a traveling path mechanism such that improving the MCV efficiency and minimizing queuing storage, this study proposes an energy threshold distribution in which nodes of *importance* are assigned higher threshold values(i.e., priority) to be charged earlier. To assign energy-threshold values, an $O(! \log !)$ online (dynamic) assign-threshold algorithm is proposed. Moreover, an adaptable charging-time distribution based on the traffic intensity and node's *importance* is promoted to fully or partially charge nodes aiming at minimizing the dead nodes density. The *importance-identification* algorithm is proposed to make the nodes identify their workload in advance.
3) To find the shortest Hamiltonian charging path in hexagonal-based topology, an efficient heuristic algorithm applied to each sector aiming at improving MCV traveling efficiency is presented. To avoid back-forth problem occurred in the MCVs due to preemption-based scheduling, the study presents Non-preemptive dynamic priority scheduling algorithm for MCVs and a centralized assignment greedy algorithm in the BS for charging tasks scheduling and assignments.
4) To the best of our knowledge, we are the first who consider the importance of nodes and develop an *importance-identification* algorithm which is a useful figure of merit in WRSN that gives us a clear insight about the important nodes those who carry heavy workload in the network that can help optimize the performance and efficiency of the network. Moreover, we are the first who develop an online energy-threshold distribution algorithm which used to configure the threshold without the need for manual setting of the energy threshold. Additionally, the ability to configure the threshold without manual intervention could save time and resources for network engineers.

The rest of this paper is organized as follows. Section 2 briefly reviews the literature. Section 3 presents the preliminaries behind this work. Section 4 explains our proposed methods. The experiments and discussions are explained in Section 5. Finally, Section 7 concludes this work. Note that this work has a supplementary file, and the figures, tables, and algorithms with underlined labels are cross-referenced from the supplementary file.

2 Related Work

Charging scheduling schemes can be classified broadly based on three criteria: *system architecture*, *charging trajectories accessibility*, and the *amount of energy to be transferred*. Due to space constraints, the related work has been moved to the supplementary file.

Unlike the aforementioned studies, we consider the impact of the network topology on the scheduling, path planning, and network lifetime. Moreover, the *importance* of the nodes indicates their influential effect on the network. For future network growth, an adaptable network topology, dynamic scheduling algorithms, and online path planning algorithms are urgent and not addressed in the aforementioned related studies. Throughout this work, we

propose a hexagonal-clustered-based topology and analytical model for nodes deployment and network planning through well-defined algorithms and equations. For optimal path planning, such that a shortest Hamiltonian path, each cluster is sliced into multiple sectors through an $O(K::N_s)$ algorithm, then for each Hamiltonian path an online energy threshold assignment algorithm is proposed to preserve the path. The *importance* of a node gives us a clear insight about its influential impact on the network, to have prior knowledge about overloaded nodes *importance-identification* with $O(!log!)$ algorithm is proposed. Finally, charging tasks assignments and further scheduling algorithms are implemented in BS and MCVs respectively to provide hybrid centralized-distributed scheduling schemes.

3 PRELIMINARIES

Network model and problem statement are the main preliminaries for constructing the algorithms proposed in this work. Note that the most frequent notations or symbols used in this article are given in TABLE 2 in the supplementary file.

3.1 Network Model

The network is modeled as an indirect graph $G_s = (S;E)$, consisting of two sets S and E. The vertices set S denoting stationary sensor nodes distributed over a region of area A. The edges set E denotes the connections between the nodes. Nodes are embedded with a Wireless Power Transfer (WPT) receiver and a rechargeable battery. There is a fixed base station BS for data collection, charging tasks assignments, and depot for MCV battery recharging or replacement. We assume that there are K MCVs to handle the charge requests assigned from the BS and located in close proximity to BS. Each MCV travels at a constant speed denoted by v_{mc} to wirelessly charge the nodes. MCVs are carrying WPT transmitter, with energy transfer rate . Fig.1 shows the proposed network model in ESPP.

3.2 Problem Statement

We consider a single cluster WRSN with layers, and K MCVs, the cluster is sliced into K sector(s) denoted as $_i; 1 \quad i \quad K$, and each sector is assigned a single MCV. To improve the MCV efficiency and reduce the response time a Hamiltonian path-finding algorithm should be applied to each sector such that finding a path W_i with the objective of minimizing traveling-cost.Therefore, we formalize the path planning problem as objective optimization problem where keeping the traveling-cost as minimum as possible, Moreover, taking node's *importance* as problem constraint.

$$\arg\min \Big(\sum_{8v_i 2 W_i} Dist(v_i; v_{i+1}) \Big)$$

$$s:t: :$$

$$Dist(v_i; v_{i+1}) \quad t_r; 8v_i; v_{i+1} \; 2 \; W_i$$

$$(v_i) > (v_{i+1}); 8v_i; v_{i+1} \; 2 \; W_i \tag{1}$$

Where $Dist(v_i; v_{i+1})$ denotes the Euclidean distance between nodes $v_i; v_{i+1}$, and (v_i) denotes the *importance* of v_i. More details are explained in Section 4.2.

We consider scheduling problem of a task set $= f_1; : : : ; _n g$ consisting of n charging-tasks on a set $M = fMCV_1; : : : ; MCV_k g$ consisting of K MCVs. We assume Non-Preemptive (NP) scheduling. We consider each task has a deadline D_i which represents the time a node should be charged before D_i to avoid irreparable influence on network performance.

The worst case response time R_i of a task $_i$ is the longest possible time from the release of a charge task until it reaches the corresponding node. Thus, the task $_i$ is said to be schedulable if and only if $R_i \quad D_i$, and task set is schedulable if and only if $8iR_i \quad D_i$. Based on these assumptions, we want to determine if we can guarantee that $= f_1; : : : ; _n g$ is schedulable under these considerations. More details are explained in Section 4.6.

4 OUR PROPOSED SCHEME

The proposed system is shown in Fig.2; mainly gives an overview of this section, and is summarized by the following five steps:

1) In order to provide a strongly connected network, the hexagonal-clustered-based network topology is proposed for nodes' deployment. Algorithm 1 is used to construct a single cluster hexagonal topology, and Algorithm 2 is used to construct the connection links between the nodes taking into account the connection cost and routing. Moreover, finding a Hamiltonian path in hexagonal-based tessellations is NP-Complete [2, 4], making MCV efficiency, and response time optimizations more feasible and accessible.
2) For dynamic scheduling policy adaptation, and determining the appropriate amount of charging energy, the *importance* of each node gives us an insight about the network core nodes (nodes of essentials), the Algorithm 3 is used to make each node identify its importance which will be used later during scheduling and charging process.
3) For a given K mobile charger vehicles, each cluster is sliced into K-sectors, and a single MCV is assigned to each sector. For each sector, a charging path planning algorithm, outlined by Algorithm 4, is applied to produce a Hamiltonian charging path.
4) Since our proposed wireless charging scheme is on-demand (online), where the nodes send charge requests when their residual energy falls into a predefined threshold, maintaining the optimal charging path is infeasible due to the random arrivals of charge requests. For this reason, we propose an energy-threshold function which is applied to the Hamiltonian charging path to preserve its validity. In order to make each node identifies its threshold, Algorithm 5 is employed with time complexity of $O(!log!)$.
5) For the charging tasks scheduling, Algorithm 6 is used, which is a non-preemptive (NP) algorithm

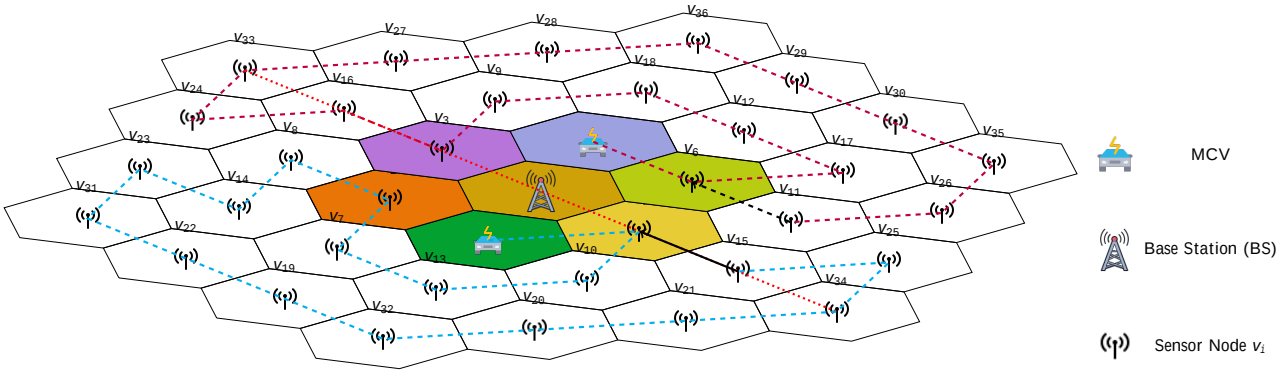

Fig. 1: Network model proposed in ESPP

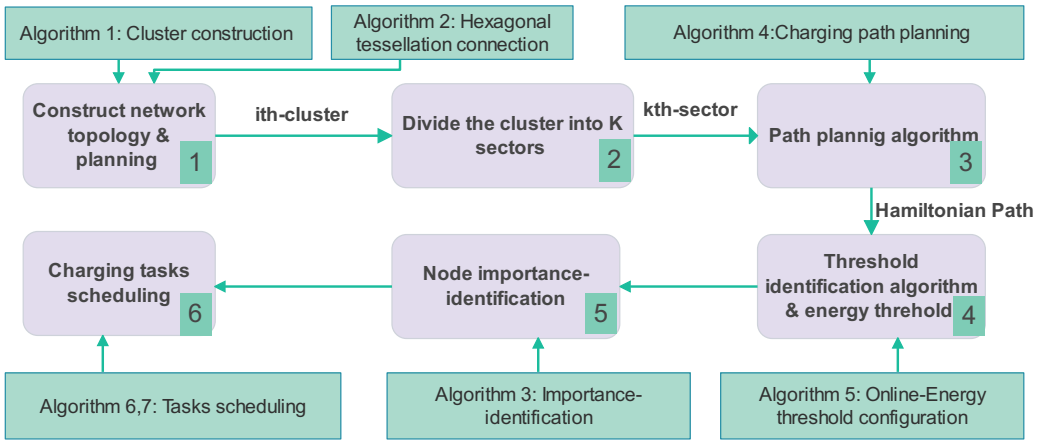

Fig. 2: The proposed system overview, Algorithm 1 and 2 are moved to Supplementary file

proposed to avoid the back-forth problem that occurs due to multiple preemptions. The BS is responsible for charging tasks assignments for MCVs based on sector ID such that the nodes in a given sector are only charged by the MCV assigned to that sector. For further scheduling and decision-making, an extra priority-based scheduling algorithm is proposed for the MCVs.

The details of these steps are presented in the following subsections.

4.1 Network Deployment & Planning

Hexagonal tessellation offers many characteristics that make it a suitable candidate for our network planning. In hexagonal tessellations, all adjacent nodes are separated by a distance of t_r, where t_r denotes the length of the edge of each regular hexagon. Each sensor node can communicate with nodes that are closer than a threshold R. Since, in a regular hexagonal tessellation, all nodes are t_r apart from their neighbors, choosing t_r R will satisfy the connectivity constraint. Hexagonal tessellation is optimal from the perspective of network coverage, as it represents the Delaunay graph of a centroidal Voronoi partition. When considering communication cost, hexagons require a minimum number of connections, three instead of six or four

We consider a farming field scenario of an area A for applications such as smart irrigation, and precision agriculture network. We assume that all nodes are homogeneous with a maximum transmission range T_r. A cluster refers to a group of nodes with a central station (i.e. BS) that are interconnected and coordinated to operate as a unit. A layer refers to a collection of nodes that are equidistant from the BS. In this context, the distance from the BS to any node in the ith layer remains constant, implying that all nodes within a given layer are positioned at the same radial distance from the BS.

The area covered by a cluster, as defined in Eq.(2) is a function of the transmission distance of the sensor node T_r and the number of layers per cluster .

$$A = \sum_{i=0}^{!} \frac{3\sqrt{3}t_r^2}{2} = (! + 1)\frac{3\sqrt{3}t_r^2}{2} \tag{2}$$

Then the number of layers that can cover a region of area A is given in Eq.(3).

$$\frac{\sqrt{\frac{8A}{3\sqrt{3}T_r^2}}}{6} \tag{3}$$

The cluster size $!$ denotes the number of nodes in the cluster

and given in Eq.(4).

$$! = 6 \sum_{i=0} i = 6\frac{(\quad + 1)}{2} = 3(\quad)(\quad + 1) = N + K + 1 \quad (4)$$

The number of sensor nodes in ith layer is given in Eq. (5).

$$N^{(i)} = 6i; 1 \quad i \quad (5)$$

The BS is initially located in the cluster center at $(x_k; y_k)$, then the location of the sensor node v_n at $(x_n; y_n)$ at row r_n, column q_n, and layer L_n in the hexagonal tessellation is given in Eq. (6).

$$\begin{aligned} x_n &= x_k + (t_r(q_n \overline{3} + \frac{\overline{3}}{2} r_n)); \quad (6) \\ y_n &= y_k + (t_r(r_n \frac{3}{2})) \\ t_r &= \frac{T_r}{\overline{3}} \\ L_n &= \sqrt{r_n^2 + q_n^2 + r_n q_n} \end{aligned}$$

The node v_n is characterized by a tuple in geometry space $(x_n; y_n; r_n; q_n; L_n; T_r)$. There are 2 rows, and 2 columns in a single cluster with layers. More details about network deployment and planning are provided in the supplementary file.

4.2 Node's Importance Identification

The *importance* of the node v_i, denoted by (v_i), is an influential metric that reflects the influential impact of a node on the network, it is related to eigenvector centrality [11] which measures the importance of a node based on the importance of its neighbors. Nodes with high eigenvector centrality are considered to be important if they are connected to other important nodes in the network [1]. The eigenvector centrality of a node is thus proportional to the sum of the eigenvector centralities of its neighbor nodes [12]. Similarly, in ESPP, the importance of the node is the sum of the importance of its neighbors. The higher the *importance* of a node is, the more important the node is in the network. Conversely, if the *importance* of a node in the network is closer to 0, that indicates the node has less contact with other nodes. Calculating the *importance* gives us a clear insight about the traffic load that a certain node will carry. This numerical value can be used for path planning and charging scheduling such that nodes with higher *importance* are assigned higher priorities to be charged prior to those with lower *importance*. Suppose that node v_i has jV_nj edges. If one of those edges is connected to node v_i, then v_i will pass on $1=jV_nj$ of its importance to v_j. The importance ranking of v_i is then the sum of all the contributions made by nodes connected to it. That is if we denote the set of nodes connected to v_i by V_n, then the *importance* $_s(v_i)$ of v_i is given in Eq. (7).

$$_s(v_i) = \frac{1}{!} \sum_{8v_j \, 2 V_n} \quad _s(v_j) \quad (7)$$

To determine the importance of a node, we first need to know the importance of all the nodes connected to it. It is a network-recursive process that requires traversing through the network and determining the importance of each node.

To do that, we propose Algorithm 3 which is a broadcast-free algorithm used to determine the *importance* of the nodes with time complexity $O(! \log !)$. Initially, the BS instantiates an *assign-importance* packet denoted as I_p according to the format $fsrc; dst; \quad ; dirg$. To maintain network addresses of adjacent nodes, we consider that addresses are temporally stored in a queue denoted as Q_n and the v_x^i denotes a parent of v_i. Each node $v_i \; 2 \; S$ selects a node v_j from its queue Q_n to which it sends I_p. When the queue is empty such that $jQ_nj = 0$, the node sends back the packet to its v_x^i. This procedure continues until traversing back to the BS which has no parent $v_x^i = ?$ which is where the algorithm terminates.

Let (i) denotes the total *importance* of the nodes in ith layer, then

$$(i) = \quad (i + 1); 0 \quad i \quad (8)$$

$$(i) = \sum_{8v_k \, 2 V_i} \quad _s(k) \quad (9)$$

For $V_i = V \; ; i = \quad ; jV \; j = 6 \quad$, then $8v_i \; 2 \; V \quad ! \quad _s(i) \; ! \; 1$. In other words, the *edge nodes* (i.e. nodes in the last layer) have average *importance* of $\frac{1}{!}$.

Algorithm 3 Node's importance-identification algorithm

```
Input: I_p packet, Q_n a queue of nodes connected to v_i
The initial importance value of v_i  _0 = 1 ;
if v_x^i = ? then
    v_x^i  I_p:src;
end if
Remove I_p:src's node from Q_n;
if I_p:dir = 1 then
     _0  I_p:  +  _0;
    (v_i)  _0;
end if
if jQ_nj = 0 then
    I_p:  _s(n);
    if v_x^i 6= ? then
        I_p:src  v_i;
        I_p:dst  v_x^i;
        I_p:dir  1;// backward
        Send I_p to v_x^i;
    end if
    Return;
end if
v_j  Dequeue node from Q_n;
I_p:dir  0;// forward
I_p:  1;
Send I_p to v_j;
```

Note that Algorithm 3 runs only once after network initialization or when topological changes occur in the network. Because after calculating the *importance* of the node, it is stored in the node (e.g in its EEPROM or in any non-volatile storage). More detailed illustrations are in the supplementary file.

4.3 Sector-based assignment & Charging Path Planning

Charging Path planning plays a critical role in WRSN since it's not only used to improve the MCVs' efficiency but also can provide high responsivity (low response time) which accordingly increases the charging throughput, minimizes the dead nodes density, and prolongs the network lifetime. MCVs usually take much time to travel for charging the nodes due to their limited speed. Since our topology is hexagonal-based tessellation, finding a Hamiltonian path is NP-Complete [8] we make use of this inherent topology Hamiltonicity property to improve the MCV efficiency and reduce the response time. In ESPP, a sector refers to a group of nodes within a specific angular range such as ($_s$; $_e$). In a sector-based assignment, the cluster is partitioned into K sectors, and each MCV is assigned to a single sector aiming at optimizing resource allocation.

Let $= f\ _1;:::;\ _K g$ denote a set of sectors in the cluster. Each $_i\ 2$ is characterized by ($_s$; $_e$) which calculated using Eq.(10).

$$\begin{aligned} {}^{i}_{s} &= \frac{2\ (i)}{K} \\ {}^{i}_{e} &= \frac{2\ (i+1)}{K} \end{aligned} \tag{10}$$

The BS is located in $(x_k; y_k)$, the angle between v_n located in $(x_n; y_n)$ and the BS is given in Eq.(11)

$$(BS; v_n) = tan\ ^{1}(\frac{y_k\ \ y_n}{x_k\ \ x_n}) \tag{11}$$

A node v_i belongs to a sector $_i$ if and only if $^{i}_{s}\ \ (BS; v_i)\ \ ^{i}_{e}$ and can only be charged by an MCV assigned to that sector such that both have a common *sector_id*. In order to minimize the response time and improve the MCV efficiency, a heuristic path planning algorithm that yields a feasible path p such that $min\ \ ^{P}_{8v_i 2p} dist(v_i; v_{i+1})$ is essential. For hexagonal-tessellation, a path planning algorithm considers the angle of view in which how the standing person in close proximity to the BS views the sector. Mainly, there are three types of paths diagonal-based, row-based, and column-based. The path is diagonal if and only if ($^{i}_{s}$ + $^{i}_{e}$) mod 2 < , and the path is said to be row-based if ($^{i}_{s}$ + $^{i}_{e}$) mod 2 = otherwise the path is column-based, path type is given in Eq.(12). Algorithm 4 produces a Hamiltonian path for each sector in which the MCV travels to each node exactly once starting from a node with higher *importance* and terminating at a node with lower *importance* (e.g. any nodes in last layers) with avoiding edge-intersection (Eulerian path). Fig.4 of the supplementary shows a single cluster with five layers and six sectors $K = 6$, and a path planning algorithm is applied to each sector.

$$p_{type}(\ ^{i}_{s};\ ^{i}_{e}) = \begin{cases} diagonal\text{-}based & (\ ^{i}_{s} + \ ^{i}_{e})\ \ \text{mod}\ 2 < \\ row\text{-}based & (\ ^{i}_{s} + \ ^{i}_{e})\ \ \text{mod}\ 2 = \\ column\text{-}based & (\ ^{i}_{s} + \ ^{i}_{e})\ \ \text{mod}\ 2 > \end{cases} \tag{12}$$

For a cluster with layer and K sector, The number of nodes in a sector is given in Eq.(13).

$$N_s = \frac{3(\ \ + 1)(\ \ )}{K} \tag{13}$$

The area covered by a sector is given in Eq.(14)

$$A_s = \sum_{i=0}^{N_s} \frac{3\sqrt{3}t_r^2}{2} = (N_s + 1)\frac{3\sqrt{3}t_r^2}{2} \tag{14}$$

The set of nodes in sector $_i\ V_i^{(\ )} = fv_1;:::;v_n g$, then $V_{\ i}$ a charging-path planning algorithm given in Algorithm 4 applied.

Algorithm 4 Charging path planning algorithm

1: **Input:** S, K;
2: **Output:** $W^0 = fw_1^0;:::;w_k^0 g$;
3: **for** $k = 1$ to K **do**
4: w_k^0 ?;
5: p_{type} Call Eq.(12)($^{i}_{s}$; $^{i}_{e}$);
6: **for** $j = 0$ to 4 **do**
7: w_j^0 ?;
8: **if** p_{type} = *diagonal-based* **then**
9: w_j^0 $8v_n\ 2\ S : (r_n + q_n = j)\ \ \&\ \ _{v_n} = i$;
10: w_j^0 order w_j^0 by r_n;
11: **else if** p_{type} = *row-based* **then**
12: w_j^0 $8v_n\ 2\ S : r_n = j\ \ \&\ \ _{v_n} = i$;
13: w_j^0 order w_j^0 by q_n;
14: **else if** p_{type} = *column-based* **then**
15: w_j^0 $8v_n\ 2\ S : q_n = j\ \ \&\ \ _{v_n} = i$;
16: w_j^0 order w_j^0 by r_n;
17: **end if**
18: **if** $j\ \ \text{mod}\ 2 = 0\ \ \&\ \ jw_j^0 j \neq 0$ **then**
19: w_j^0 reverse w_j^0;
20: **end if**
21: **if** $jw_j^0 j \neq 0$ **then**
22: w_k^0 $w_k^0\ [\ w_j^0$;
23: **end if**
24: **end for**
25: W^0 $W^0\ [\ w_k^0$;
26: **end for**
27: **return** W^0

The Algorithm 4 works as the following, given a set of nodes S in a cluster, and K sector as an input. In Line 5, we calculate the azimuth angles of a sector $_i$ using Eq.(10), then figuring out the path type from Eq.(12). In Line 8, 11, and 14, checking out the path type. In Line 9, mapping $8v_n\ 2\ S : (r_n + q_n = j)\ \ \&\ \ _{v_i} = i$ nodes such that the condition $(r_n + q_n = j)\ \ \&\ \ _{v_i} = i$ is satisfied which yields a set of diagonal nodes in the sector i, the resulting map-set w_j^0 is then ordered by r_n since rows with small indices have greater *importance* , the same procedure applied to other path types except for *column based*, the map-set is ascendingly ordered by q_n as in Line 16. For inter-layer transition and node's *importance* constraint, the path is interchanged on even-layers as outlined in Line 19. The output Hamiltonian paths $w_k^0\ 2\ W^0$ satisfy that $8v_i; v_{i+1}\ 2\ w_k^0 8w_k^0\ 2\ W^0$ the $Dist(v_i; v_{j+1}) = t_r$.

In order to preserve the charging path for on-demand charging scheme for each sector, a threshold function $\Gamma(w_k^0)$ is applied to the set $w_k^0\ \ 2\ W^0$, then $8v_i\ 2\ w_k^0$ an assign-

threshold function is applied such $ff_{th} : i ! e^i_{th}8v_i 2 w^0_k;0 \quad i \quad jw^0_k jg$ using Algorithm 5. After applying the threshold kernel function to the path, the resulting energy threshold set $fe^{(1;k)}_{th};:::;e^{(n;k)}_{th}g$ ensures that $e^{(1;k)}_{th} > e^{(2;k)}_{th} > ::: > e^{(n;k)}_{th};8w^0_k 2 W^0$. Intuitively, if $e^{(i)}_{th} > e^{(j)}_{th}$ and $(v_i) > (v_j)$, then v_i will send a charge request before v_j.

We assume the assign-threshold kernel function F_{th} as in Eq.(15).

$$F_{th}(i;k) = \ _0 + \ _1 \exp \ \frac{1}{2}(\frac{i}{jw^0_k j})^2 \tag{15}$$

Where $_0; \ _1$ and are kernel parameters. Then the energy threshold of v_i in sector $_k$ is given in Eq.(16).

$$e^{(n;k)}_{th} = e_{max}F_{th}(n;k); \ _0 \quad F_{th}(n;k) \quad _1 \tag{16}$$

4.3.1 Path Planning Correctness Proof

In order to prove the correctness of the path planning algorithm and preserve the global charging path in ESPP, let a Hamiltonian path $P^{(k)}_h = fv_1;:::;v_ng$ in sector $_k$, after applying the threshold algorithm, the energy threshold for the nodes follows the pattern $e^{(1;k)}_{th} > e^{(2;k)}_{th} > ::: > e^{(n;k)}_{th}$, and importance of the nodes such that $(v_1) > \quad > (v_n)$, $8v_i 2 P^{(k)}_h$ sends charge request in the sequence $v_1 ::: v_n$. The arrival time follows the sequence $t_a(i;k) < ::: < t_a(n;k)$, and the waiting time $t_q(i;k) > ::: > t_q(n;k)$, if nodes $fv_i;v_j;:::;v_mg$ in the charging queue, the $t_q(v_i) > t_q(v_j) > \quad > t_q(v_m)$, then according to the assigned priority in Eq.(39) to each corresponding charging task, the MCV will schedule the tasks and travel through the path $fv_i;v_j;:::;v_mg$ which intuitively by contradiction proves path preservation for on-demand charging scheme.

4.4 Energy Threshold

The Energy threshold value of the node e_{th} determines at what time a node v_i will send a charge request. It plays a critical role in the charging process and other charging factors. The Algorithm 5 optimizes the ESPP algorithm by efficiently traversing the network to configure energy thresholds for nodes based on a predetermined distribution such as Eq.(16). This results in three main benefits:

TABLE 1: ASSIGN THRESHOLD PACKET FORMAT

src	*dst*	*type*	n	N_s		$_0$	$_1$

Reduced Queue Length: Intelligent configuration of energy thresholds ensures a more even distribution of nodes sending charge requests, leading to shorter queues and reduced waiting times at the MCVs. *Decreased Miss Rate of Charge Requests:* An even distribution of energy thresholds minimizes the chances of multiple nodes reaching the energy threshold at once, ensuring fewer missed charge requests due to congestion and timely energy provisioning. *Time Efficiency:* The Algorithm's structured approach avoids the resource-intensive broadcasting method, saving time and network resources in configuring energy thresholds. Ultimately, these enhancements increase the efficiency and reliability of the ESPP algorithm by allowing the network to respond more dynamically and effectively to the energy needs of individual nodes.

Algorithm 5 Online-Energy threshold configuration algorithm

Input: $type = 1; N_s = jw_k j; \ ; n = 0; \ _0; \ _1;$
initialize the algorithm in the BS
packet Format packet based on Table 1;
v_{first} Dequeue node from Q_n;
packet:src BS;
packet:dst v_{first};
Send *packet* to v_{first};

In the receiving node v_i **or** BS;
Input: *packet*;
Extract input fields from *packet*;
n *packet:n* field;
if *flag* 6= 1 **then**
  Set threshold based on Eq. (16);
  v^i_x *packet:src*;
  flag 1;
  packet:n = n + 1;
end if
Remove *packet:src* node from Q_n if exists;
if $jQ_n j = 0$ **then**
  if *parent* 6= ? **then**
    Send *packet* to v^i_x;
  end if
  Return;
end if
v_{next} Dequeue node from the Q_n;
packet:src v_i;
packet:dst v_{next};
Send *packet* to v_{next};

4.5 Charging Model with Multiple MCVs

In ESPP on-demand charging scheme, sensor nodes continuously keep track of their residual energy e^n_r when the energy reaches a configured threshold e^n_{th} such $e^n_r \quad e^n_{th}8v_n 2 S$, a charge request is sent to the BS for pre-scheduling and assignments. In the BS, the collected requests are distributed over K MCV based on *sector_id*. The charge request packet is based on the format in Table.2 containing its location $(x_n;y_n)$, residual energy $e^n_r \quad e^n_{th}$, estimated energy consumption rate E_c, and its *importance* $_s$ is used for estimating charging duration, priority, and other useful information. The *sector_id* of the v_n is obtained using Eq.(10), and Eq.(11).In ESPP, the charging is a hybrid charging scheme in which the charging duration depends on the traffic(e.g. number of charge requests in the queue of a certain MCV) and the importance of the node. Let $w^0_k = fv_1;v_2;:::;v_ig$ be a set of sensor nodes in sector $_k$, where $jw^0_k j$ denotes the number of sensor nodes in that sector. Each $v_n 2 w^0_k$ has an

TABLE 2: CHARGE REQUEST PACKET FORMAT

x_n	y_n	e^n_r	E^n_c	$_s$

energy threshold value $e_{th}^{(n;k)}$ given by Eq.(16). The residual energy $e_r^{(n;k)}$ of v_n at a given time t is given by Eq.(17).

$$e_r^{(n;k)} = e_{max} \quad tE_c \tag{17}$$

Without loss of generality, a node $v_n \ 2 \ w_k^{\emptyset}$ will send a charge request when $e_r^{(n;k)} = e_{th}^{(n;k)}$, then the Eq.(17) is rewritten in Eq.(18), where $t_{req}(n;k)$ denotes the charge request time of node v_n in sector k.

$$e_{th}^{(n;k)} = e_{max} \quad t_{req}(n;k)E_c^n \tag{18}$$

We assume negligible transmission time such that the request arrival time is $t_a(n;k) = t_{req}(n;k)$. Substituting the $e_{th}^{(n;k)}$ given in Eq.(16) in Eq.(18), then the arrival time distribution of charge requests is given in Eq.(19), where $_1 = (\frac{1}{E_c^n}{}^{\emptyset})$, and $_2 = \frac{1}{E_c^n}$.

$$\begin{aligned} t_a(n;k) &= \frac{e_{max}^n \quad e_{th}^{(n;k)}}{E_c^n} \\ &= \frac{e_{max}^n(1 \quad F_{th}(n;k))}{E_c^n} \\ &= e_{max} \quad _1 \quad _2 \exp \quad \frac{1}{2}(\frac{n}{jw_k^{\emptyset}j})^2 \end{aligned} \tag{19}$$

Suppose if successive nodes demand a charge service at epochs $:::; t_a(n;k); t_a(n+1;k); :::;$ and if $u(n)$ denotes the inter-arrival time $t_a(n+1;k) \quad t_a(n;k)$, then the random variables $::; u(n); u(n+1); :::$ are statistically independent and enjoy the same arbitrary distribution $\frac{@t_a(n;k)}{@n}$ [5]. Then the inter-arrival time distribution of charge requests is given in Eq. (20).

$$u(n;k) = \frac{e_{max} \ _2 n}{(\ jw_k^{\emptyset}j)^2} \exp \quad \frac{1}{2}(\frac{n}{jw_k^{\emptyset}j})^2 \tag{20}$$

The average inter-arrival time of charge request denoted as is given in Eq. (21).

$$_k = \quad \frac{e_{max} \ _2}{(\ jw_k^{\emptyset}j)^3} \sum_{n=0}^{jw_k^{\emptyset}j} n \exp \quad \frac{1}{2}(\frac{n}{jw_k^{\emptyset}j})^2 \tag{21}$$

The average arrival rate of charge requests, denoted by $\hat{}_a(k)$, is the reciprocal of the average inter-arrival time (k). In other words, it is the expected number of charging requests that arrive per unit time, from the sector, $_k$ is given in Eq. (22) and measured in request per second, and the arrival rate of charge requests in the cluster of K sector is given in Eq.(23).

$$\hat{}_a(k) = \frac{1}{k} \tag{22}$$

$$\hat{}_a = \sum_{k=1}^{K} \frac{1}{k} = \frac{K}{k} \tag{23}$$

The arrival rate $\hat{}$ of charge requests at a given time t is given in Eq.(24).

$$\hat{}(t) = \frac{t}{k} \tag{24}$$

The busiest-hour time t_{max} is defined as the time in which the queue length of a given MCV will reach its maximum value given in Eq.(25).

$$t_{max}(k) = \ _2 e_{max} \quad 1 \quad \exp \quad \frac{1}{2}(\frac{jw_k^{\emptyset}j \quad 1}{jw_k^{\emptyset}j})^2 \tag{25}$$

Assume that, a set $V_r \quad S = fv_1; v_2; :::; v_r g$ of requesting nodes. Each node $v_n \ 2 \ V_r$ sent a charge request, these requests are sector-based distributed through an assignment algorithm over a set $M = fMCV_1; MCV_2; :::; MCV_k g$ of MCVs. Each MCV estimates the residual energy of each node according to Eq.(26).

$$e_r^{\emptyset n} = e_r^n \quad (t_q(n)E_c) = e_{th}^n \quad (t_q(n)E_c) \tag{26}$$

We consider a non-preemptive on-demand charging scheme where each $MCV_k \ 2 \ M$ will continue to charge the node without interruption. This may cause energy depletion for some nodes, especially in dense networks. To mitigate that, a Traffic-based Partial Charging scheme (TPC) is proposed which considers the workload of the corresponding MCV with a *charging parameter* denoted as which can be tuned to balance the average energy efficiency vs dead nodes density. Unlike, preemption-based partial charging where lower priority tasks can be preempted when higher priority tasks arrived causing back-forth problems for the MCV. In ESPP, preemption is prohibited and the charging time is adaptable according to node's *importance* $_s(n)$, the estimated average inter-arrival rate of charge request per each sector $_k$ and the queue length . The time required to charge $v_n \ 2 \ S$ when available charge requests is given in Eq.(27).

$$T_{charge}(n; \ ) = \quad \frac{e_{max} \quad e_r^{\emptyset n}}{!} \quad e \tag{27}$$

$$= \frac{1}{1 + 3: \ _q \log \ (k)} \tag{28}$$

$$_q = \sum_{8i2Q} \ _s(i) \tag{29}$$

Where Q denotes the charging queue. The maximum energy capacity $E_{max}^{MC_k}$ of $MCV_k \ 2 \ M$ must satisfy condition in Eq.(30), assuming that the energy consumed in traveling given as E_c^{mc} measured in $J=m$. Due to the hexagonal-tessellation, the Hamiltonian path produced by Algorithm 4, and the priority assigned to each task, then $8v_n \ 2 \ v_r$ the distance between $D(v_n; v_{n+1}) = D(MCV_k; v_n) = t_r$.

$$\begin{aligned} E_{max}^{MC_k}(\ ) \quad & \sum_{n=0}^{X} (E_{travel}^n + \ :T_{charge}(n; \quad n)); \\ & \sum_{n=0}^{X} ((n+1)t_r E_c^{mc}) + \sum_{n=0}^{X} e_{max} e^{\ (\quad n)} \\ & \frac{(\ +1)(\ +2)t_r E_c^{mc}}{2} + \frac{e_{max}(1 \quad e^{\ (\ +1)})}{1 \quad e} \end{aligned} \tag{30}$$

where E_{travel}^n is the energy needed to travel to v_n.

$$_{mcv} = \lim_{! \ 0} E_{max}^{MC_k}(\ ) = (\ +1)e_{max} + \frac{(\ +1)(\ +2)t_r E_c^{mc}}{2} \tag{31}$$

$$^{\emptyset}_{mcv} = \lim_{! \ 1} E_{max}^{MC_k}(\ ) = e_{max} + \frac{(\ +1)(\ +2)t_r E_c^{mc}}{2} \tag{32}$$

The estimated average energy obtained by all nodes can be

calculated by Eq.(33).

$$E(\ ) = \frac{}{+1} \sum_{n=0}^{} T_{charge}(n; \quad n) \tag{33}$$
$$= \frac{e_{max}}{+1} \frac{(1 \quad e^{\ (\ +1)})}{1 \quad e}$$

The service time $t_s(n)$ of a given charge request from v_n is given as $t_s(n) = T^n_{travel} + T_{charge}(n; \ )$, and the total service time t_s required to complete a task set of size is given in Eq.(34).

$$t_s(\ ) = \sum_{n=0}^{} (T^n_{travel} + T_{charge}(n; \quad n)) \tag{34}$$
$$= \sum_{n=0}^{} \frac{(n+1)tr}{v_{mc}} + \sum_{n=0}^{} \frac{e_{max}}{} e^{\ (\ \ n)}$$
$$= \frac{t_r(\ +1)(\ +2)}{2v_{mc}} + \frac{e_{max}}{} \frac{(1 \quad e^{\ (\ +1)})}{1 \quad e} \Bigg!$$

$$_{ts} = \lim_{\ ! \ 0} t_s(\ ) = \frac{(\ +1)e_{max}}{} + \frac{t_r(\ +1)(\ +2)}{2v_{mc}} \tag{35}$$

$$^{0}_{ts} = \lim_{\ ! \ 1} t_s(\ ) = \frac{e_{max}}{} + \frac{t_r(\ +1)(\ +2)}{2v_{mc}} \tag{36}$$

The nth arriving request must wait $t_q(n)$ in the queue to obtain the service, the waiting time of two successive charge requests is given in Eq. (37).

$$t_q(n+1; \ ) = t_q(n; \quad 1) + t_s(n; \quad 1) \quad u(n) \tag{37}$$

and $u(n)$ is the inter-arrival time between nth and $(n+1)th$ charge request.

4.6 Charging Tasks Scheduling in ESPP

Scheduling is the process of assigning charging tasks over a set of MCVs. The charging scheduling framework proposed in ESPP is depicted in Fig.5 of the supplementary file. Let $M = fMCV_1; : : : ; MCV_k g$ denote mobile charge vehicles, and $= f\ _1; : : : ;\ _n g$ denotes a set of charging tasks arrived in the BS. Each $MCV_k \ 2 \ M$ is assigned to a single sector denoted as $_i$. We consider that each $_i \ 2$ is characterized by $(v_i; status_i; T_i; \ _i; R_i; D_i; C_i)$ tuple where v_i denotes the requesting node, $status_i$ denotes the status of task, T_i is the arrival time of $_i$, $_i$ denotes the sector which v_i belongs to, R_i is the response time, D_i is the deadline of the task, and $C_i = T^i_{travel} + T_{charge}$ is the worst case service time. The task $_i$ is said to be schedulable if and only if $R_i \quad D_i$, and task set is schedulable if and only if $8 \ _i \ 2 \quad R_i \quad D_i$.

The task has four statuses and changed in each phase, 1) *created* the task $_i$ is said to be created when it is inserted into the queue, the creation time (arrival time) $t_a(i) = T_i$. 2) *assigned* the task $_i$ is assigned when the $_i$ is assigned to the MCV such that $^{MCV}_i = \ _i$, $t_x(i)$ denotes the assigned time. 3) *released* when the MCV starts the execution of the task by traveling to S_i, $t_{rel}(i)$ denotes the release time of $_i$. 4) *completed* when the MCV completes the execution of the task (charging + traveling), $t_{comp}(i) = C_i$ denotes the time the task $_i$ completes. The assignment Algorithm 6 is used to assign charging tasks to the MCVs, The algorithm

Algorithm 6 Assign charging tasks algorithm implemented in BS

1: **Input:** $; M;$
2: **for Each** $MCV_k \ 2 \ M$ **do**
3: $\quad _t \quad 8 \ _i \ 2 \ : \ _i = \ ^{MCV_k}_i;$
4: $\quad$ **for Each** $\ _i \ 2 \ \ _t$ **do**
5: $\quad\quad$ **if** $\ _i$ *is schedulable with* $MCV_k \quad \& \quad _i$*:status* ~~$6=$~~ *assigned* **then**
6: $\quad\quad\quad$ Assign $\ _i$ to $MCV_k;$
7: $\quad\quad\quad$ Update $\ _i$ status *assigned;*
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad$ Release tasks assigned to $MCV_k;$
11: **end for**

6 works as follows, given two sets: the tasks set , and a set of MCVs M as inputs. In order to maintain the charging path the algorithm group the tasks as in line 3 such that tasks with the same *sector_id* compared to MCV *sector_id* are grouped together. In line 5, check if the task $_i$ is schedulable with MCV_k such that $R_i \quad D_i$. In lines 6,7, assign the schedulable tasks to their corresponding server(MCV) and update their status respectively.

Algorithm 7 Scheduling algorithm proposed in MCV.

1: **Input**: a charging tasks set;
2: ^{0} a priority queue;
3: **for Each** $\ _i \ 2 \quad$ **do**
4: $\quad P_i \quad$ Calculate the initial priority from Eq.(38);
5: $\quad ^{0} \quad ^{0} \ [\ f(\ _i; p_i)g;$
6: **end for**
7: **while** ^{0} ~~$6=$~~ ? **do**
8: $\quad \hat{}_i \quad$ Select the task with highest priority from $^{0};$
9: $\quad \hat{v}_i \quad \hat{}_i{:}v_i;$
10: $\quad$ Travel to the $\hat{v}_i;$
11: $\quad$ Calculate the estimated charging time from Eq.(27);
12: $\quad$ Charge $\hat{v}_i;$
13: $\quad$ Notify BS that $\hat{}_i$ has completed;
14: $\quad$ Update tasks' priorities based on Eq.(39);
15: **end while**

The Algorithm 7 is a type of scheduling algorithm based on dynamic priority scheduling in which the priorities are initially calculated during the task assignment and updated after any task completion. It aims to adapt to dynamically changing progress and to form an optimal configuration in a self-sustained manner. Moreover, directing the scheduling policy to an actively plausible path which intuitively improves the MCV efficiency. In the algorithm, the tasks assigned from BS are enqueued in a priority queue ^{0} in the MCV. In Lines 3-6, a priority is initially assigned to each task and calculated from Eq.(38) forming a set of charge tasks denoted as $^{0} = f(\ ^{0}_1; P_1); : : : ; (\ ^{0}_n; P_n)g$. According to the scheduling policy, the task with highest $\hat{}_i$ is served prior to those with lower priority as outlined in Line 8. Each MCV_k will travel to charge the node (denoted as $\hat{v}_i$) with the corresponding highest task $\hat{}_i$. The time required to travel to a node $\hat{v}_i$ is obtained by $\hat{t}_{travel} = \frac{Dist(MCV_k; \hat{v}_i)}{v_{mc}}$,

where v_{mc} denotes the travelling velocity of corresponding MCV. The charging time with $= j\ ^{0}j$ requests is given in Eq.(27).

$$P_n = \frac{(n)}{D(MC_k; v_n)} \tag{38}$$

When a charging task is completed, the MCV will send a notification packet to the BS to alter $\wedge_i$ status as in Line 13. The MCV will update the priorities associated with each task in the priority queue as outlined in Line 14 such that $8(\ _i; P_i)\ 2\ \ ^{0}\ !\ 8(\ _i; P_i^{0})\ 2\ \ ^{0}$. The new priorities are calculated from Eq.(39).

$$P_n^{0} = \frac{t_q(n)\ \ (n)}{E_r^{0n} D(MC_k; v_n)} \tag{39}$$

The response time of a task $_n$ denoted as R_n is the possible time from the release of the charge task until the time the MCV arrives at the corresponding node. Thus, the utilization of a charge task $_n$ denoted as U_n is given in Eq.(40), and the total utilization of a charge task set is given in Eq.(41).

$$U_n = \frac{C_n}{u(n)} = \frac{T_{travel}^{n} + T_{charge}(n;\ )}{u(n)} \tag{40}$$

$$U = \frac{1}{K} \sum_{n=0}^{X} \frac{C_n}{u(n)} = \frac{1}{K} \sum_{n=0}^{X} \frac{T_{travel}^{n} + T_{charge}(n;\ \ \ n)}{u(n)} \tag{41}$$

4.7 Algorithms Complexity

The time complexity of the node's importance-identification algorithm and Online-Energy threshold configuration algorithm in Algorithm 3, 5 is $O(!\ \ \log !)$. While it's $O(K\ \ 4\ \ !\log !)$ for path planning Algorithm 4 More details of the proposed algorithms' time complexity analysis are in the supplementary file.

5 Simulation and Results

This section is devoted to the performance evaluation of our proposed ESPP scheme. Extensive experimental simulations have been performed. Our simulation code can be found on [1].The simulation parameters are listed in Table 3. More details on the simulation are provided in the supplementary file.

5.1 Experimental Results

We compared our proposed ESPP scheme with both mTS [9] and NPDC [13] in terms of queue length, response time, average energy, and the density of dead nodes.

5.1.1 Queue Length

The queue length refers to the number of charge requests (or tasks) waiting in an MCV's queue for charging service. The queue length can be measured in terms of the number of tasks or the amount of time that a charging task spends waiting in the charging queue. It is an important metric in WRSN as it helps to determine the performance of a queuing system in the network. If the queue length is too long, it may indicate that the system is not able to handle the incoming charging requests efficiently, which can result in long waiting time and a decrease in the network lifetime. Queue length is affected by various parameters as the arrival time distribution of charge requests, and the service time. From Eq.(19), we notice that the arrival time of charge requests depends on the energy threshold function F_{th} parameters $(\ _0;\ _1;\ ;jw_kj)$, and the energy consumption E_c. And from Eq.(34), we notice that service time depends on the MCV speed v_{mc} and charging time parameter . Unlike mTS, our scheme ESPP introduces a mechanism to control the arrival rate of charge requests using the energy threshold distribution. Unlike both NPDC and mTS, our scheme ESPP solves the MCV speed constraint by introducing a path planning algorithm, both of these solutions make the our scheme ESPP performing better in terms of queue length. From Fig. 3(a), we notice that ESPP achieves a much shorter queue length than those of mTS and NPDC because of the two solutions proposed in ESPP to minimize the queue length. We conclude that ESPP has promising results in terms of queue length which increase the overall charging efficiency.

TABLE 3: SIMULATION PARAMETERS

Parameter	Value
Number of layers	13
Number of sensor nodes N	546
Number of MCVs K	10
Number of sectors	10
Sensor node energy (e_{max})	100J
Energy consumption of v_n E_c^n	0.01-0.04J/s
Mobile charger Energy E_{mc}	12KJ
Mobile charger speed	5m/s
Sampling time	160 320 sec
Energy per packet E_{packet}	0:001J=packet
	0.45
^{0}	5%
^{1}	35%
T_r	50m

5.1.2 Average Energy

The average energy efficiency is the average energy obtained after charging the nodes in the network. There are many parameters that affect the average energy such as charging rate , MCV's speed v_{mc}, charging duration, scheduling algorithm, and path planning policy. In ESPP, the average energy depends on the traffic which adapts the charging time to minimize the dead nodes' density.

From Eq.(33), we notice that in order to maximize the average energy, minimizing the queue length by controlling the arrival time of charge requests is essential. As previously mentioned in Subsection 5.1.1, ESPP outperforms mTS and NPDC in terms of queue length due to the proposed solutions which have not only a positive impact on the queue length but also on the average energy.

Fig.3(b) shows the energy efficiency of ESPP is more stable and increases with time. The average energy efficiency of ESPP is higher than that of NPDC and mTS. An overall intuitive reason is not only due to the path

1. https://github.com/NanoSoft774849/WRSN

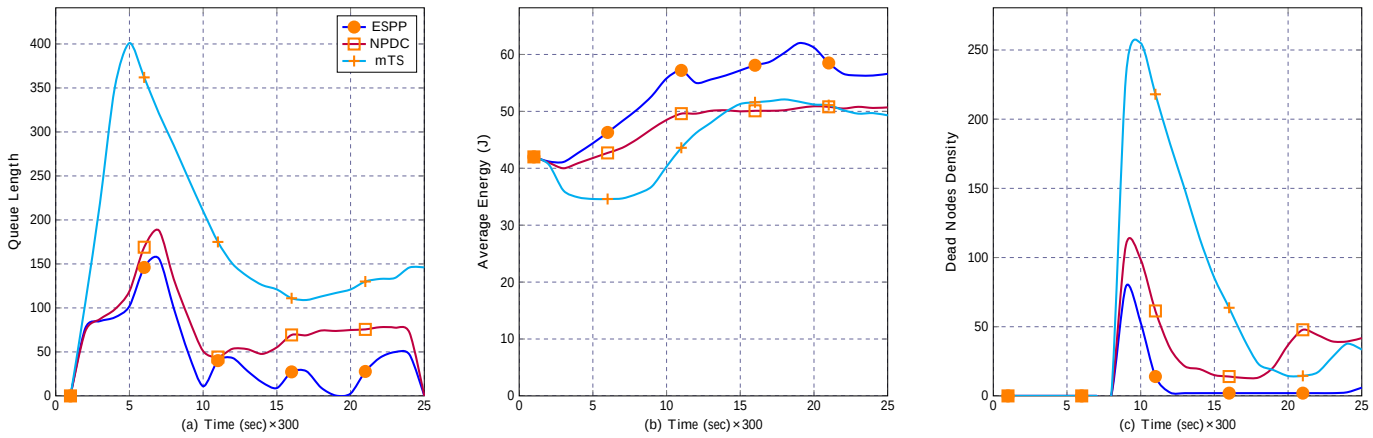

Fig. 3: (a) Total queue length, (b) Average energy, and (c) Dead nodes density

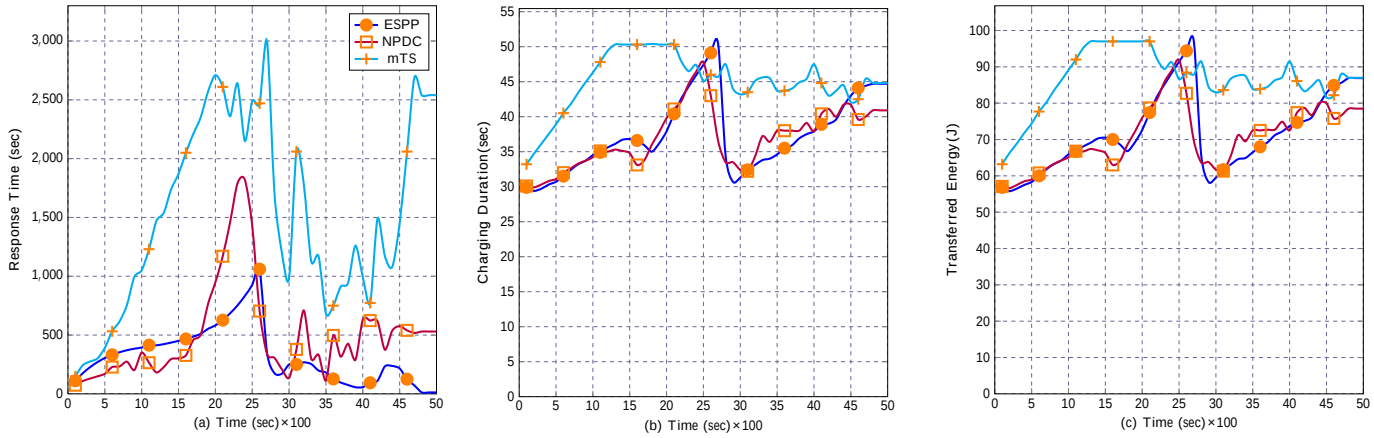

Fig. 4: (a) Response time, (b) Charge duration, and (c) Obtained energy

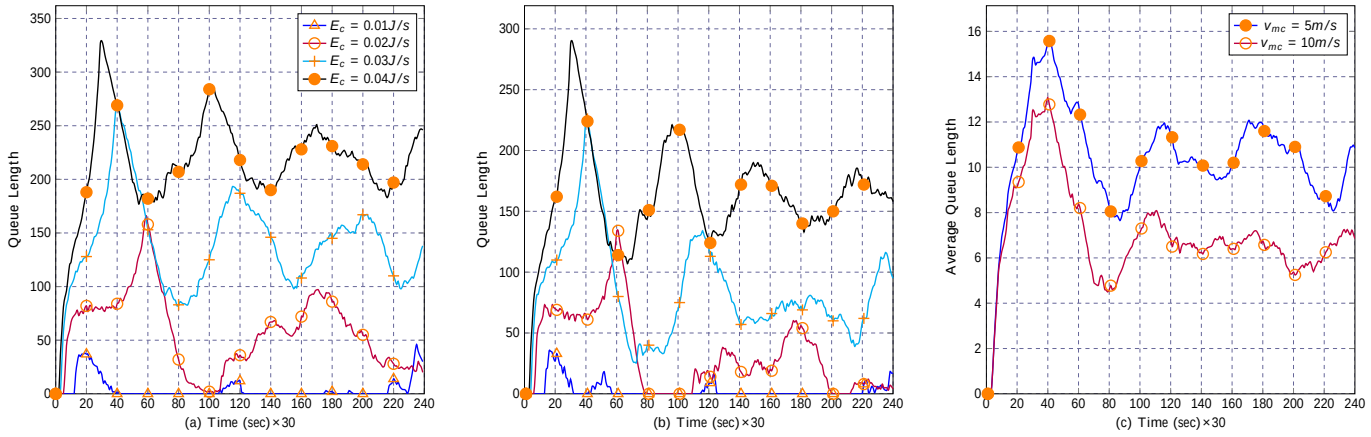

Fig. 5: (a) The queue length at v_{mc} = 5m=s, (b) The queue length at v_{mc} = 10m=s, and (c) The average queue length at different speeds

planning and scheduling algorithms but also due to the energy threshold distribution proposed in ESPP. Hence, we conclude that ESPP achieves a promising average energy efficiency extending the lifetime of the network.

5.1.3 *Density of Dead Nodes*

The density of dead nodes indicates the number of exhausted nodes during a given time duration. Fig.3(c), during the busiest hour time the density of dead nodes of three algorithms are relatively high due to the configured residual energy at the simulation startup. Then, gradually, the density of the ESPP algorithm converges to 0 while others the density fluctuates between 20-50. The density of ESPP is always lower than those of NPDC and mTS. The highest density of ESPP, NPDC, and mTS are 14.65%, 20%, and 45.78%, respectively. We conclude that ESPP can ensure

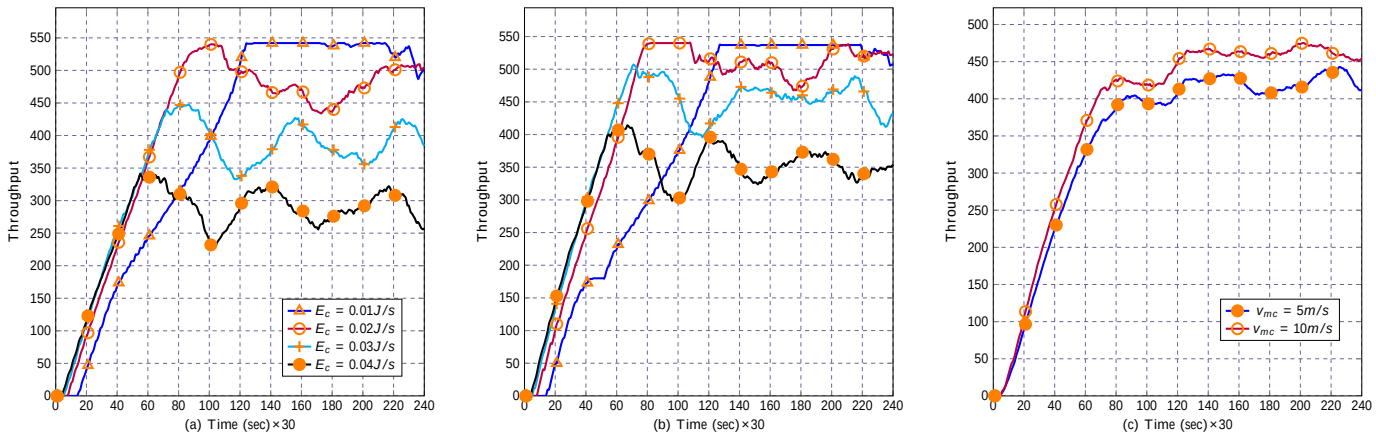

Fig. 6: (a) The throughput at v_{mc} = 5m=s, (b) The throughput at v_{mc} = 10m=s, and (c) The throughput at different speeds

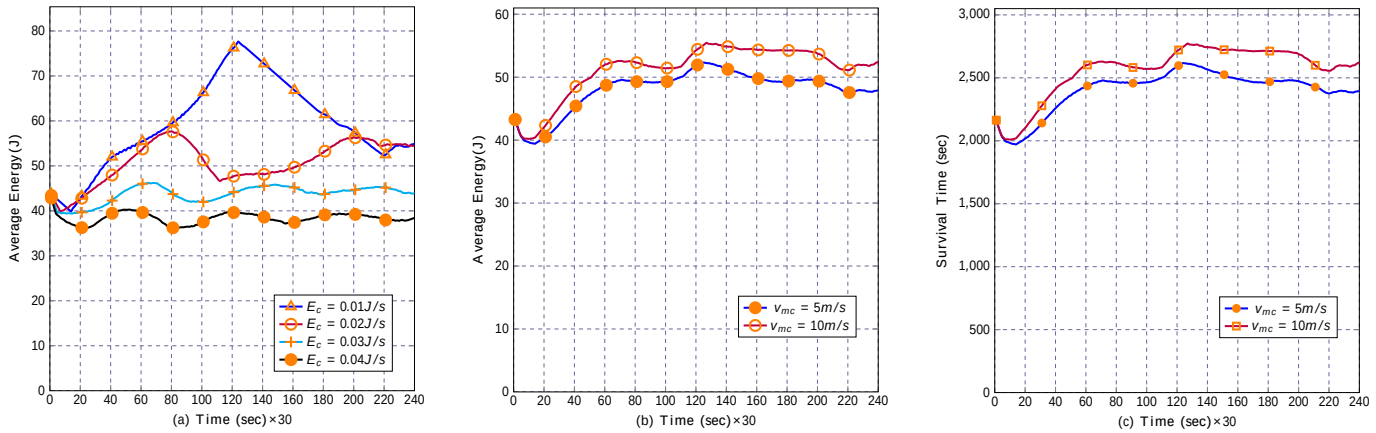

Fig. 7: (a) The average Energy at v_{mc} = 5m=s, (b) The average energy at different speeds and different energy consumption, and (c) Survival time

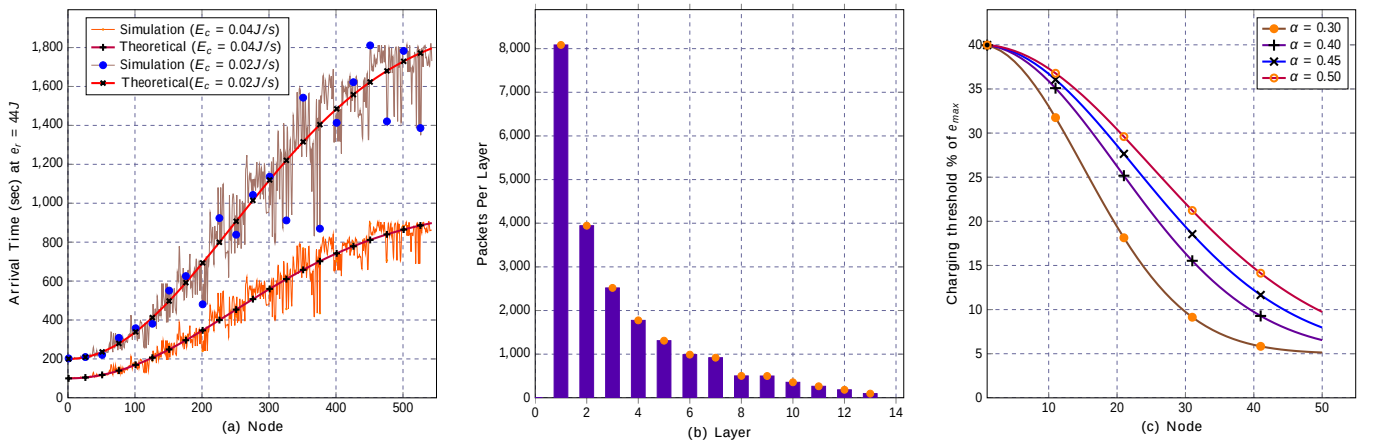

Fig. 8: (a) Theoretical Vs simulation arrival time, (b) Traffic per layer, and (c) Charging threshold at different values of

fewer dead nodes, which is beneficial for long-term stable operations of the network, and our objective is achieved.

5.1.4 *Average Response Time*

The response time refers to a time interval that starts when MCV is assigned a charge request from the BS and ends when the MCV arrives at the corresponding node. As previously mentioned, the traveling path of ESPP is the shortest Hamiltonian path; intuitively, ESPP will respond to requests very quickly. As shown in Fig. 4(a), at first, an energy threshold function is applied to the path to gain control over charging requests arrivals. In the beginning, the response time of mTS algorithm is long due to the massive arrival of charge requests. We note that ESPP performs a faster response time. The reason is that, in ESPP, the path planning is well-designed to minimize the response time.

We conclude that ESPP gives a promising response time far better than mTS and NPDC.

5.1.5 Theoretical Vs Simulation Arrival Time

The arrival time (AT) of charge requests depends on the energy threshold distribution and energy consumption rate. Fig.8(a) shows ESPP arrival time distribution from the simulation and the theoretical arrival time distribution. We notice that the analytical arrival time distribution in Eq.(19) depends on the energy threshold distribution in Eq.(16). The Mean Squared Error (MSE) between max-normalized theoretical AT and max-normalized simulation AT is 0.00431841, 0.004426536 for $E_c = 0{:}04J/s$ and $E_c = 0{:}02J/s$ respectively. The variations in the simulation arrival time shown in Fig.8(a) are due to that, the nodes in the simulation network send data packets to BS randomly between (160-320) sec, this randomness imposes different energy consumption which makes some nodes send charge requests before others leading to slight variations in arrival curve. We conclude that the analytical arrival time highly correlates with the simulation arrival time.

5.1.6 Charging Throughput or Charging Tasks Throughput

The charging throughput denotes the number of charged nodes during a given period of time. In other words, it refers to the number of charging tasks completed within a specific time period. It is a measure of the efficiency of the system. Figs.6(a),(b) show the charging throughput at different energy consumption and different MCV speed. Fig.6(c) shows the average throughput at different E_c and v_{mc}. In these figures, we notice that the charging throughput is greatly affected by the MCV speed and energy consumption.

5.1.7 Impact of Low Eficient Nodes

Usually, low-efficient nodes have a great impact on the charging efficiency [10]. In ESPP, the nodes in the last layers are usually assigned smaller energy thresholds which intuitively means that they will lately send charge requests that minimize their impact on the charging efficiency. Moreover, these nodes have low Eigenvector centrality meaning that less influence on the network lifetime. Fig.8(b) illustrates the network packets that are either sent or forwarded by the nodes. It is observable that the number of packets diminishes as one ascends to the higher layers of the network.

5.1.8 Charging Threshold

The energy threshold dictates the point in time at which a node will send a charge request. In the ESPP scheme, we employed an energy threshold distribution as defined in Eq. (16) and used a network-traverse Algorithm 5 to assign the energy threshold. Fig. 8(c) illustrates the energy threshold distribution utilized in ESPP. The distribution is intuitively derived from the traffic load across various layers, as depicted in Fig. 8(b), and is distributed at the sector level to ensure that nodes that impose higher traffic are assigned a correspondingly higher energy threshold while nodes with less contact are assigned lower energy threshold to mitigate the impact of low-efficient nodes during charging scheduling. Overall, the charging threshold has a great influence on the charge request arrival rate and consequently on the queue length and charging process.

5.1.9 Energy Consumption

The energy consumption of the node is a manufacturer-related parameter, that depends on various aspects such as the MPU/MCU type, the type of sensors used, and their ADCs. The energy consumption E_c of the nodes has a great impact on the charging system such as queue length, throughput, and average energy. Fig.5(a,b) shows queue length at different energy consumption rates and different speeds respectively. Fig.7(a) shows average network energy when different E_cs are used. Fig.6(a,b) shows the charging throughput with different E_c. The reason is that when nodes consume more energy they send more charge requests which lead to higher queue length and lower throughput which in turn affect the network lifetime. One prominent solution is to use well-designed power management scheduling protocols, such protocols allow the node to enter different power states, such as sleep or idle, when not in use. This reduces power consumption and helps to prolong the network lifetime.

5.1.10 Speed Impact of MCV

The speed of the MCV has a great impact on many system parameters including queue length, charging throughput, and response time. Fig.5(c) illustrates the impact of speed on the average queue length(averaged over different multiple energy consumption E_c and all the MCVs such that $\frac{1}{4:K}\sum_{i=0}^{K} {}_i\cdot$) which shows that the higher the speed the lower the queue length because when the speed is high, the response time will be smaller and the service rate will be high which means a smaller number of nodes (charge tasks) will be waiting in the queue to be charged. Fig.6(c) shows the impact of the speed on the charging throughput, the higher the speed the higher the throughput, that is due to the fact that the MCVs will need less time to arrive at the nodes. Furthermore, we noticed that average network energy increased when the MCVs travel at higher speeds as shown in Fig.7(b). Overall, the speed of the MCVs has a great influence on many results and higher speeds promise less dead nodes' density and higher network lifetime.

5.1.11 Survival Time

The survival time or the average survival time is a time measure of how much a network will survive before the nodes get exhausted. The average survival time is related to the average obtained energy at a given time divided by energy consumption. Fig.7(c) shows the network survival time at different MCV speeds. We conclude that network survival time is affected by many parameters such as energy consumption, MCV speed, scheduling, and path planning algorithms.

6 Node Design

To prove the feasibility of this work, we have provided the "Node Design" Section in the supplementary file. We have concluded that this work and its proposed algorithms can be implemented in real-world wireless rechargeable sensor networks.

7 CONCLUSIONS

In this paper, we have proposed an on-demand charging scheme for charging sensor nodes in WRSN along with a hexagonal-clustered-based network topology for nodes' deployment and network planning. In order to improve MCV efficiency, we first propose an online importance-identification algorithm to calculate the importance of each node in the network which gives us a clear insight into their influence in the network which we used in path planning and scheduling. We further propose a heuristic algorithm that produces a Hamiltonian charging path for each sector and then applies an energy threshold function to that path using an online algorithm to maintain the path and gain control over the charge requests' arrival rate. Furthermore, we propose a priority-based scheduling algorithm for task scheduling and assignment. The accuracy of the analytical results and the efficiency of the proposed system design have been verified through extensive simulations. Future ongoing research is exploring the practical implementation and deployment challenges associated with using UAVs to access a variety of risky environments. Additionally, conducting more extensive comparisons with learning-based algorithms, as well as validating the proposed algorithms through random distribution of nodes, will be essential parts of our future work to enhance the generalizability of the results.

ACKNOWLEDGMENTS

This paper is supported by Innovation Team and Talents Cultivation Program of National Administration of Traditional Chinese Medicine (No. ZYYCXTD-D-202208). Al-Dubai and Hussain also acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) Grants Ref. No. EP/T021063/1 and EP/T024917/1. Saeed work emanated from research conducted with the financial support of Science Foundation Ireland under Grant number SFI/12/RC/2289_P2.

REFERENCES

[1] Walid Ahmad, Mason A. Porter, and Mariano Beguerisse-Díaz. "Tie-Decay Networks in Continuous Time and Eigenvector-Based Centralities". In: *IEEE Transactions on Network Science and Engineering* 8.2 (2021), pp. 1759–1771. DOI: 10.1109/TNSE.2021.3071429.

[2] Esther M Arkin et al. "Not being (super) thin or solid is hard: A study of grid Hamiltonicity". In: *Computational Geometry* 42.6-7 (2009), pp. 582–605.

[3] Liang He et al. "Evaluating the On-Demand Mobile Charging in Wireless Sensor Networks". In: *IEEE Transactions on Mobile Computing* 14.9 (2015), pp. 1861–1875. DOI: 10.1109/TMC.2014.2368557.

[4] Kaiying Hou and Jayson Lynch. "The computational complexity of finding hamiltonian cycles in grid graphs of semiregular tessellations". In: *arXiv preprint arXiv:1805.03192* (2018).

[5] David G. Kendall. "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain". In: *The Annals of Mathematical Statistics* 24.3 (1953), pp. 338–354. DOI: 10.1214/aoms/1177728975. URL: https://doi.org/10.1214/aoms/1177728975.

[6] Manoj Kumar, Sushil Kumar, and Pankaj Kashyap. "Effect of Harvesting Unpredictability of resources in Energy Harvesting-WSN". In: *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*. 2021, pp. 1–5. DOI: 10.1109/GUCON50781.2021.9573729.

[7] André Kurs et al. "Wireless power transfer via strongly coupled magnetic resonances". In: *Science* 317 (5834 2007). ISSN: 00368075. DOI: 10.1126/science.1143254.

[8] Klavdija Kutnar and Dragan Marušič. "Hamilton cycles and paths in vertex-transitive graphs—current directions". In: *Discrete mathematics* 309.17 (2009), pp. 5491–5500.

[9] Chi Lin et al. "mTS: Temporal-and spatial-collaborative charging for wireless rechargeable sensor networks with multiple vehicles". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 99–107.

[10] Chi Lin et al. "TSCA: A Temporal-Spatial Real-Time Charging Scheduling Algorithm for On-Demand Architecture in Wireless Rechargeable Sensor Networks". In: *IEEE Transactions on Mobile Computing* 17.1 (2018), pp. 211–224. DOI: 10.1109/TMC.2017.2703094.

[11] Laishui Lv et al. "Eigenvector Centrality Measure Based on Node Similarity for Multilayer and Temporal Networks". In: *IEEE Access* 7 (2019), pp. 115725–115733. DOI: 10.1109/ACCESS.2019.2936217.

[12] Matheus R. F. Mendonça, André M. S. Barreto, and Artur Ziviani. "Approximating Network Centrality Measures Using Node Embedding and Machine Learning". In: *IEEE Transactions on Network Science and Engineering* 8.1 (2021), pp. 220–230. DOI: 10.1109/TNSE.2020.3035352.

[13] Abdulbary Naji, XingFu Wang, and Ammar Hawbani. "Efficient Non-preemptive On-demand Charging Scheduling Scheme for WRSN". In: Institute of Electrical and Electronics Engineers Inc., 2022, pp. 311–320.

[14] Muhammad Umar Farooq Qaisar et al. "SDORP: SDN based Opportunistic Routing for Asynchronous Wireless Sensor Networks". In: *IEEE Transactions on Mobile Computing* (2022), pp. 1–1. DOI: 10.1109/TMC.2022.3158695.

[15] Xianpeng Qiao et al. "A Novel Magnetically Coupled Resonant Wireless Power Transfer Technique Used in Rotary Ultrasonic Machining Process". In: *2021 IEEE Wireless Power Transfer Conference (WPTC)*. 2021, pp. 1–4. DOI: 10.1109/WPTC51349.2021.9457864.

[16] Madhvi Saxena and Subrata Dutta. "Improved the efficiency of IoT in agriculture by introduction optimum energy harvesting in WSN". In: *2020 International Conference on Innovative Trends in Information Technology*