

# Chapter 4

## MS-based data collection in heterogeneous networks

### 4.1 Introduction

The Internet of Things (IoT) is revolutionizing the way we interact with the world around us. It is enabling the development of many new technologies and applications. Wireless Sensor Networks (WSNs) are a primary component of any IoT-based system [21]. In WSNs, Sensor Nodes (SNs) are used to collect data from the monitoring environment and transmit it wirelessly to the Base Station (BS) [74]. The potential applications of WSNs are vast and varied. In the agricultural sector [115], WSNs are used to monitor soil moisture levels and temperature, which enables farmers to optimize their irrigation systems. SNs used in the WSN are low-cost, small, and have an in-built battery inside them. In WSNs, data collection and transmission operations are the main reasons for the battery's depletion during sensing cycles [116]. After the deployment, it is not always possible to replenish or change the battery of SNs due to the harsh environment [117]. Therefore, the short life span is one of the major challenging issues in WSNs. Furthermore, most WSNs use multi-hop communication for data transmission due to the limited communication range of SNs. SNs near the BS

transmit more packets than faraway SNs. This leads to fast depletion of the battery in SNs near BS. They die early as compared to faraway SNs [46]. The early death of the SNs significantly increases data collection delay due to the unavailability of efficient data routing paths in the network. The early death of the SNs also divides the network into different isolated sub-networks. These sub-networks are unable to communicate individually to the BS due to the lack of a data routing path. It is subject to the premature death of the network. This problem is also known as the hot-spot problem or sinkhole problem [51].

A Mobile Sink (MS) provides an effective solution to this problem. The MS is a movable wireless device that moves within the WSN and performs data collection [84]. It prolongs the network lifetime by reducing the energy consumption of SNs [108]. Rendezvous Points (RPs) are locations in the network where MS halts to collect data from SNs. Existing MS-based data-gathering approaches select dedicated RPs within the network for data collection from SNs. In the existing MS-based approaches, the number of RPs is very high. It significantly increases the data collection delay. Existing MS-based approaches are also unable to identify the shortest data-gathering path due to improper RP selection within the networks [62, 64]. Existing approaches do not take into account the SN buffer space while selecting RPs. It causes buffer overflow at SNs and leads to data loss. Furthermore, most of the existing approaches primarily focus on data gathering in homogeneous WSNs [36, 37, 38]. The heterogeneous WSNs present a complex network environment compared to homogeneous WSNs. In heterogeneous WSNs, sensor nodes exhibit varied energy capacities, communication ranges, and buffer sizes. The data-gathering approaches designed for homogeneous WSNs are unable to adapt to heterogeneous WSNs. The above issues motivate us to propose an MS based data collection scheme for heterogeneous WSNs. On the other hand, if any node in the network buffers excess packets, then the data routing delay also becomes very high. It is known as bufferbloat problem. The existing approaches are also unable to handle

bufferbloat problems.

This chapter proposes an MS-based data-gathering scheme for IoT-enabled heterogeneous WSNs. A novel Energy Efficient Rendezvous Points Selection (EERPS) algorithm is proposed to select the optimal number of RP. It also identifies an optimal number of Cluster Heads (CHs) and forms clusters that prevent premature death of the SNs. The proposed scheme helps to avoid network partitioning. The proposed scheme considers the different buffer capacities of SNs during the CH selection within the network. It helps to avoid the data loss due to buffer overflow in SNs. It also enables the effective utilization of network resources and improves overall network performance. Furthermore, this work applies a deep learning algorithm known as Deep Policy Dynamic Programming (DPDP) to form an optimal path for MS. To find an optimal path, DPDP shows better performance as compared to the classical Travelling Salesperson Problem (TSP) and other neural network algorithms [118]. Thus, the MS can gather data with minimal delay and transmit it to the BS. It significantly reduces data collection delay and prevents the bufferbloat problem. The proposed Energy Efficient Rendezvous Points Selection algorithm and Deep Policy Dynamic Programming based optimal path selection algorithm together referred as the EERPS-DPDP approach throughout the thesis. The following are the main contributions of this chapter.

1. This chapter presents an EERPS algorithm that identifies an optimal number of CHs and creates optimal clusters in the IoT-enabled heterogeneous WSNs. The proposed approach selects CHs based on the varying communication ranges, buffer sizes, and battery capacities of the SNs. Furthermore, it also identifies an optimal number of RPs within the network. It helps to improve network lifetime and prevent network partitioning.
2. This chapter proposes a DPDP algorithm to identify an optimal path for the MS. The DPDP based algorithm significantly reduces data collection delay and improves the overall performance of the network.

**Table 4.1:** Terminologies and definitions

<i>Terminologies</i>	<i>Definition</i>
$\mathcal{N}$	Number of SNs.
$\mathcal{S}$	Set of SNs. $\mathcal{S} = s_1, s_2, s_3, s_4, \dots, s_N$ .
$p$	Number of rendezvous points.
$RP$	Set of rendezvous points. $RP = R_1, R_2, R_3, \dots, R_p$ .
$E_t(b, d)$	Energy required to transmit $b$ bits of data over the distance $d$
$E_r(b)$	Energy required to receive $b$ bits of data.
$\mathcal{E}_{ij}$	Edge between node $i$ and $j$ .
$\mathcal{EC}$	Energy Consumption.
$\mathcal{DCD}$	Data Collection Delay.
$\mathcal{DLBO}$	Data Lost Due to Buffer Overflow.
$InDeg_i$	In-degree of a node $i$ .
$\delta(R_i, R_j)$	Euclidean distance between node $R_i$ and $R_j$

- Extensive simulations and test-bed experiment results show that the proposed approach outperforms the existing state-of-the-art approaches in terms of network lifetime, network stability, data loss due to buffer overflow, residual energy, and delay.

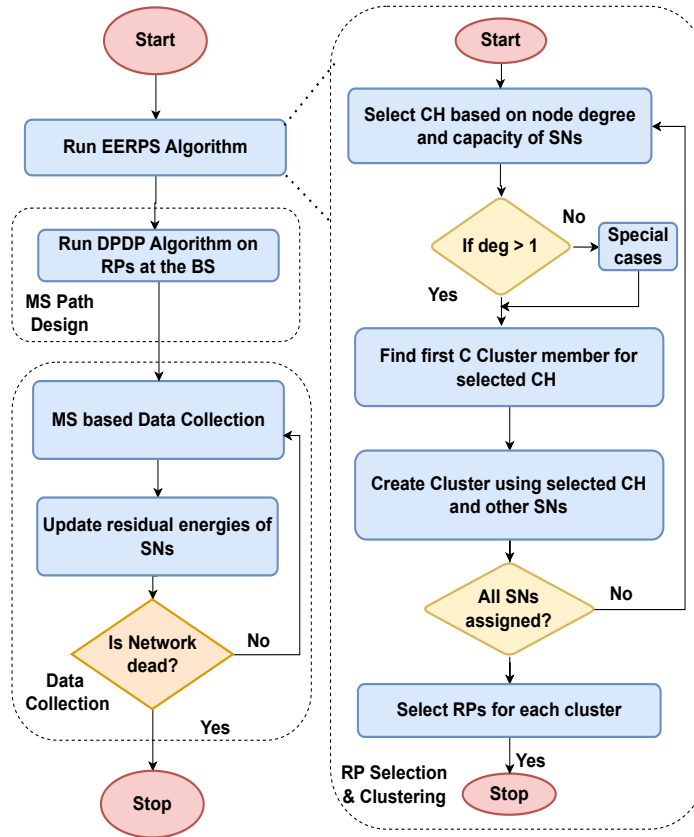
## 4.2 Network Model

In the proposed EERPS-DPDP scheme,  $\mathcal{N}$  number of SNs are deployed randomly in a  $\mathcal{A} \times \mathcal{A} [m^2]$  region. The deployed SNs are static and heterogeneous. The SNs have varied energy capacities, communication ranges, and buffer sizes. There are three types of SNs such as low-capacity, medium-capacity, and high-capacity. High-capacity SNs have high energy capacity, extended buffer size, and high communication range. It helps to prevent data loss and bufferbloat problems. Medium-capacity SNs have higher capacity compared to low-capacity SNs but lower capacity compared to high-capacity SNs. Low-capacity SNs have low energy capacity, small buffer size, and low communication range. An MS is used to gather the sensed data in WSNs. SNs send their data to CH, and CH forwards this data to MS when MS reaches the nearest RP. MS relays the gathered data to BS in each sensing cycle. The MS moves at a constant speed and has sufficient storage and computational capacity. An SN is considered a dead node if the remaining energy of the SN is zero. The transmission distance between the two SNs or between

the SN and CH is the Euclidean distance between them. In this work, the WSN is represented as a directed graph  $\mathcal{G} = \{\mathcal{S}, \mathcal{E}\}$ .  $\mathcal{S}$  is the set of SNs, and  $\mathcal{E}$  is the set of edges in the graph. Two nodes in the graph have a directed edge only if both nodes are within the communication ranges of each other. Table 4.1 contains the terminology and definition of the variables that are used in this chapter. A first-order radio model similar to [65] is used in this chapter to calculate the energy consumption of the SNs.

### 4.3 Proposed Scheme

The proposed EERPS-DPDP scheme is divided into three phases such as the optimal number of CH and RP selection algorithm, the deep learning-based optimal path design mechanism for MS, and the MS-based optimal data collection mechanism. In the first phase, the optimal CHs are selected using the proposed EERPS algorithm. It minimizes the total hop counts and significantly minimizes the energy expenditure of SNs. Furthermore, the proposed EERPS algorithm selects an optimal number of RPs for data collection from CHs. The proposed EERPS algorithm considers SNs' varying buffer sizes, energy capacities, and communication ranges during CH selection. In the second part, the DPDP algorithm is used to obtain an optimal path of MS traversal. DPDP is a deep learning algorithm, and it effectively computes the MS path as compared to other traditional TSP and neural network algorithms. The optimal path design algorithm minimizes the distance travelled by the MS while visiting all RPs. It significantly reduces data collection delay. Fig. 4.1 shows the architecture of the proposed EERPS-DPDP scheme. The proposed EERPS-DPDP approach is executed at the BS. In the third phase, MS visits RPs in the network and collects data from the CHs. MS receives the optimal path information from BS and starts data collection. MS offloads its data to the BS at the end of each data collection tour. The algorithms are executed at the beginning of the network operations. Furthermore, if the energy of a CH goes below a predefined threshold or any CH becomes faulty, these algorithms

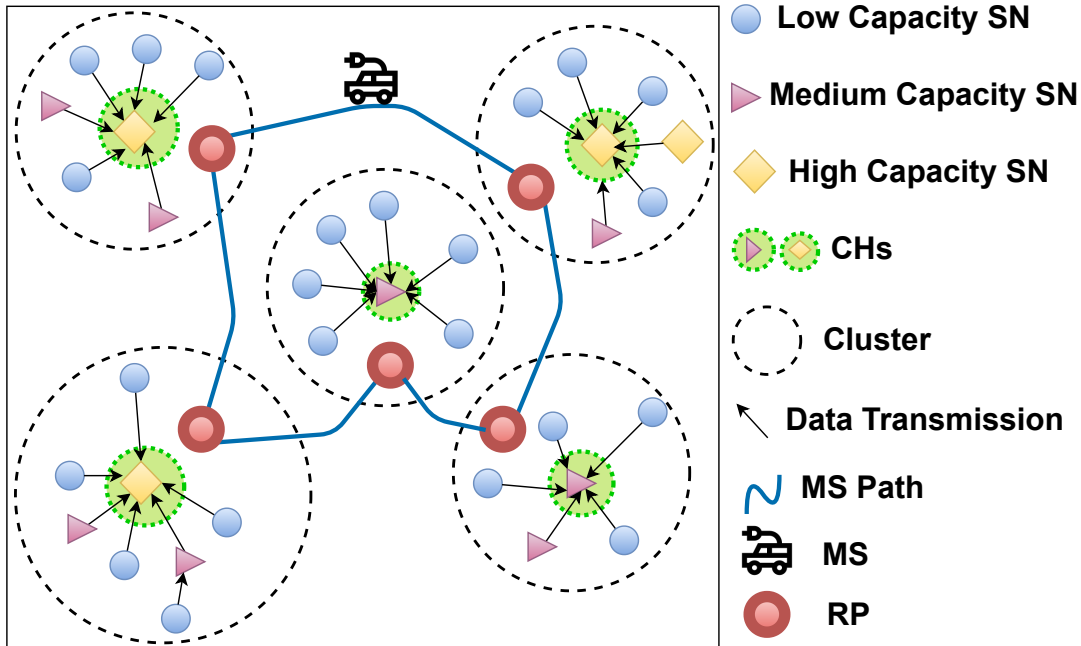


**Figure 4.1:** Architecture of the proposed EERPS-DPDP approach.

are triggered again.

#### 4.3.1 Optimal number of cluster head and rendezvous point selection

The proposed EERPS algorithm is used to select the optimal number of CHs in the network. In the heterogeneous network, the high-capacity or medium-capacity SNs are selected as parent nodes for isolated SNs. Isolated SNs are those nodes that are unreachable from CH. The capacity of the parent SN is greater or equal to the child SN. A parent node relays not only its own data but the data of its child node. Hence, SNs with higher energy capacity and higher communication range bear the load of data transmission. It helps to save energy at low-capacity SNs. It balances energy consumption in the cluster. Furthermore, it improves SN's lifetime, which is also helpful



**Figure 4.2:** MS based data collection with optimal number of cluster and RPs.

in enhancing the overall network lifetime. SNs with high in-degree are selected as CHs. It minimizes the hop counts required for cluster member SNs to send data to CH. It enables uniform energy consumption throughout the cluster. Thus, it maximizes the stability period and network lifetime. Once CH is selected, the RP for this cluster is selected within the CH communication range. In the proposed scheme, RPs are selected in such a way that MS does not collide with CHs during data collection. Otherwise, CHs will be damaged physically. Furthermore, it also helps to minimize the data collection delay. Fig. 4.2 shows the MS-based data collection mechanism with an optimal number of CHs and RPs. The pseudo-code of the proposed EERPS algorithm is shown in algorithm 2. Table 4.2 shows the terminologies used in algorithm 2.

The EERPS algorithm creates a directed adjacency list of SNs in the network. These SNs are sorted in descending order based on their capacity and stored in a list  $\rho$ . A parent array  $\mathbb{P}$  is initialized to store the parent of each SN. Initially, all nodes are considered as potential CHs. The following steps are repeated for all SNs in  $\rho$ . The SN

**Table 4.2:** Algorithm 2 terminologies

<i>Terminologies</i>	<i>Definition</i>
$\mathbb{P}$	Set containing every node's parent
$\mathbb{C}$	Capacities of the SNs
$\mathcal{D}$	List of Communication range of all SNs
$C$	Maximum number of SNs in a initial cluster
$\rho$	Potential CHs List
$\mathcal{E}$	Adjacency list of the created graph of SNs
$ngb_i$	List of nodes for which $\mathcal{E}_{ij} \neq 0$ , where $j = 1, 2, 3, \dots, n$
$minClustNode_i$	Selects a node from $ngb_i$ which is in a cluster with minimum density
$maxClustNode_i$	Selects a node from $ngb_i$ which is in a cluster with maximum density
$argSort(k)$	Return list of indices of k which would sort k in ascending order

with the highest in-degree is selected as the current CH, which is referred to as  $crp$  in algorithm 2. The in-degree of SNs is calculated with the remaining nodes in the  $\rho$ . The SNs connected to  $crp$  are selected as cluster members. An SN joins a cluster if only the capacity of the SN is less than or equal to the CH. If the in-degree of  $crp$  is greater than or equal to a threshold  $C$ , then the first  $C$  number of SNs from its adjacency list are selected as its cluster members.  $\mathbb{P}$  and  $\rho$  are updated accordingly. For instance, if *node A* has an in-degree greater or equal to  $C$ , then the first  $C$  number of nodes that are in  $\rho$  and have an edge with *node A* are included in the same cluster. In this cluster, *node A* acts as the CH. It equally distributes the load among CHs for balanced energy consumption. Furthermore, all the nodes from the cluster are removed from the  $\rho$  list.

There are two special cases where the in-degree of  $crp$  is either one or zero. If the in-degree of  $crp$  is one (e.g., *node A* has an edge with *node B*) then a node not present in  $\rho$  (e.g., *node X*) is selected from the cluster with the maximum density in such a way that it has an edge from *node A* i.e.  $maxClustNode_a$ . Furthermore, these three nodes form a cluster. The *node A* is declared as a CH. Second, if the in-degree of  $crp$  is zero for a node (e.g., *node C*) then a node is selected in such a way that it has an edge over *node C*. It is present in the cluster with minimum density, i.e.  $minClustNode_c$ . Then, *node C* becomes the child of this selected node and part of its cluster. *Node C* uses two hops to communicate with the CH via the selected node.  $\mathbb{P}$  and  $\rho$  are updated accordingly. These steps are repeated until the  $\rho$  list is empty. Fig. 4.3a and Fig. 4.3b show the first special case, and Fig. 4.3c and Fig. 4.3d show the second special case. In

**Algorithm 2:** EERPS Algorithm

---

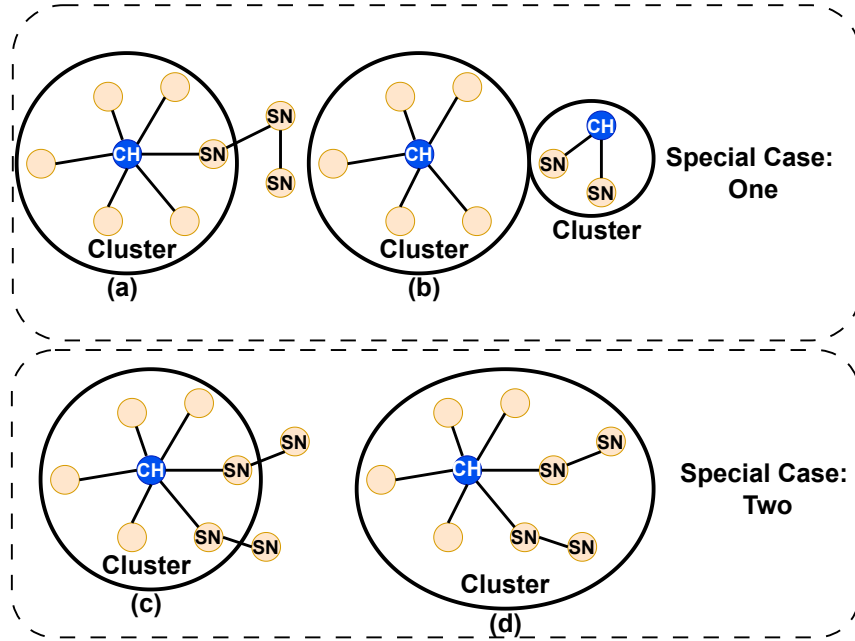
**Input:**  $\mathcal{S}, \mathcal{C}, \mathcal{D}, \mathcal{C}$   
**Output:** Parent of each SN,  $\mathbb{P} = [\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n]$ , Set of RPs  $RP$

- 1 **Initialize:** Adjacency list  $\mathcal{E}$ ,  $\rho = reverse(argSort(\mathcal{C}))$ ,  
 $\mathbb{P} = [-1, -1, \dots, n \text{ times}]$  **while**  $\rho \neq \Phi$  **do**
- 2     highdeg  $\leftarrow$  -1, crp  $\leftarrow$  -1
- 3     **for** node  $s_i$  in  $\rho$  **do**
- 4         rpd  $\leftarrow$  sizeof(  $\mathcal{E}[node] \cap \rho$  )
- 5         **if** rpd > highdeg **then**
- 6             crp  $\leftarrow$   $s_i$ , highdeg  $\leftarrow$  rpd
- 7         **if** highdeg  $\geq \mathcal{C}$  **then**
- 8             **break**
- 9     **if** highdeg==0 **then**
- 10         For node  $s_i$ ,  $a = minClustNode_i$
- 11         **if**  $a \neq \phi$  **then**
- 12              $\mathbb{P}[i] = a$ , update  $\rho$
- 13         **else**
- 14             crp  $\leftarrow$   $s_i$
- 15     **if** highdeg==1 **then**
- 16         For node  $s_i$ ,  $a = maxClustNode_i$
- 17          $\mathbb{P}[a] = s_i$
- 18         crp  $\leftarrow$   $s_i$ , update  $\rho$
- 19      $\rho \leftarrow \rho - crp$
- 20      $\mathbb{P}[crp] \leftarrow crp$
- 21     childs  $\leftarrow$  select first  $\mathcal{C}$  nodes from  $\mathcal{E}[crp] \cap \rho$
- 22     **for** child in childs **do**
- 23          $\mathbb{P}[child] \leftarrow crp$
- 24      $\rho \leftarrow \rho - childs$
- 25 Select RPs using equation 4.3 and 4.4 and add to  $RP$
- 26 **return**  $\mathbb{P}$ ,  $RP$

---

Fig. 4.3, blue nodes represent CH, and the rest of the nodes are SNs. The black circle shows the cluster.  $Edge_{ij}$  represents the edge between nodes  $i$  and  $j$ , which is defined as follows.

$$Edge_{ij} = \begin{cases} 1, & if \delta(i, j) \leq \mathcal{D}[i] \& \delta(i, j) \leq \mathcal{D}[j] \& \mathcal{C}[i] \geq \mathcal{C}[j] \\ 0, & otherwise \end{cases} \quad (4.1)$$



**Figure 4.3:** Special cases in RP selection and clustering.

In-degree of a node  $i$  is calculated as follows.

$$InDeg_i = sizeOf(\mathcal{E}[i]) \quad (4.2)$$

In a cluster, cluster member nodes communicate with CH in either single-hop or two-hop communication. After the optimal number of cluster formations, RPs are selected within the communication range of the CHs in such a way that MS does not collide with CH during the data collection process. The location of the RPs is calculated using the following equation.

$$X_i = x_j \pm \mathcal{D}[j]/2 \quad (4.3)$$

$$Y_i = y_j \pm \mathcal{D}[j]/2 \quad (4.4)$$

where  $X_i, Y_i$  are coordinates of RP,  $x_j, y_j$  are coordinates of CH, and  $\mathcal{D}[j]$  is communication range of CH.

### 4.3.2 A deep learning-based optimal path design mechanism for MS

After selecting the RPs, the DPDP algorithm is used to identify an optimal data-gathering path for the MS traversal. A complete graph is created using the RPs coordinates. Furthermore, a Graph Neural Network (GNN) is used to calculate the probabilities (heatmap) of the edges in the graph. This heatmap is used to construct the optimal solution using the dynamic programming algorithm iteratively. The calculation of a traversal path that MS follows to visit RPs is done at the BS. The objective is to minimize the MS tour length  $\mathcal{TL}$ , which is represented as follows.

$$\mathcal{TL} = \sum_{i=1}^p \sum_{j \neq i, j=1}^p \mathcal{L}_{R_i, R_j} \cdot \delta(R_i, R_j) \quad (4.5)$$

subject to:

$$\delta(MS, CH_i) \leq \mathfrak{D}[i] \quad (4.6)$$

$$\prod_{R_i \in RP} \Gamma_{R_i} = 1 \quad (4.7)$$

$$\prod_{s_i \in S, R_j \in RP} \Upsilon_{i,j} = 1 \quad (4.8)$$

where  $\mathcal{L}_{R_i, R_j} \in [0, 1]$ , 1 indicates that the MS path includes the link  $(R_i, R_j)$ . Variable  $\Gamma_{R_i} \in [0, 1]$ , 1 means that RP  $R_i$  appears only once in the path. Variable  $\Upsilon_{i,j} \in [0, 1]$ ,  $\Upsilon_{i,j}$  is 1 if SN  $s_i$  is covered by RP  $R_j$ . Eq. 4.6 ensures that MS is within communication range of the visited CH. Eq. 4.7 ensures that each RP is visited once in a data-gathering tour. Eq. 4.8 ensures that each SN/CH is covered by at least one RP. Algorithm 3 shows the pseudo code for the proposed DPDP algorithm. Table 4.3 shows the terminologies used in algorithm 3.

#### 4.3.2.1 Deep policy dynamic programming

Each partial solution in the DPDP algorithm is defined by three parameters such as  $cost(R)$ ,  $current(R)$ , and  $visited(R)$ . The  $cost(R)$  is the total distance travelled till now,

**Table 4.3:** Algorithm 3 Terminologies

<i>Terminologies</i>	<i>Definition</i>
$B$	Beam size
$Q$	Empty beam
$R$	Empty trace
$M$	Graph neural network
$H$	Heatmap
$T$	Tour containing optimal path

---

**Algorithm 3:** DPDP based Optimal Path Design Algorithm

---

**Input:** A set of RPs  $RP$ , a beam size  $B$   
**Output:** A tour  $T$  containing optimal path

- 1 Initialize an empty beam  $Q$ , empty trace  $R$ , GNN  $M$
- 2 Evaluate  $M$  on  $RP$  to obtain heatmap  $H$
- 3 Add the initial solution  $a_0 = (s)$  to  $Q$
- 4 **for**  $t=1$  to  $|RP|$  **do**
- 5     Initialize an empty set  $E$
- 6     **for each** solution  $a \in Q$  **do**
- 7         **for each feasible action**  $u \in RP(a)$  **do**
- 8             Generate an expanded solution  $a' = (a, u)$
- 9             Update  $H(a')$
- 10            Add  $a'$  to  $E$
- 11     Remove dominated solutions from  $E$
- 12     Select the top  $B$  solutions from  $E$  according to Score and assign them to  $Q$
- 13     Store the parent and action information of each solution in  $Q$  to  $R$
- 14 Backtrack the best solution  $T$  from  $Q$  using  $R$
- 15 **return**  $T$

---

$current(R)$  is the currently visited node, and  $visited(R)$  is the set of all visited nodes. The algorithm starts with node  $R_0$  as an initial solution. Other parameters such as  $cost(R_0)$  is initialized as 0 and  $current(R_0)$ ,  $visited(R_0)$  are initialized as  $R_0$ . The whole solution is built iteratively. The transition of the parameter at state  $t$  is  $R = R_0, \dots, R_{t-1}$  to  $R' = R_0, \dots, R_t$ . The next RP is defined as follows.

$$cost(R') = cost(R) + c_{current(R), R_t} \quad (4.9)$$

$$current(R') = R_t \quad (4.10)$$

$$visited(R') = visited(R) \cup R_t \quad (4.11)$$

There can be multiple solutions for a single Dynamic Programming (DP) state, but only the minimum cost solution is assigned to the DP state. The minimum cost solution among all solutions is the one with the minimum  $cost(R)$ . The size of the solution is limited to  $O(B)$  at each state, where  $B$  is the beam size. DPDP uses a GNN to predict the heatmap  $h_{ij}$  of the edges. It denotes the probability of the edge  $e_{ij}$  to be in the optimal solution. A scoring policy is designed using this heatmap to rank the solutions. Furthermore, the top  $B$  solutions are selected for the next iteration.  $score(R)$  is calculated as follows.

$$score(R) = heat(R) + potential(R) \quad (4.12)$$

$$heat(R) = \sum_{i=1}^{t-1} h_{R_{i-1}, R_i} \quad (4.13)$$

$$potential(R) = potential_0(R) + \sum_{j \notin visited(R)} potential_i(R) \quad (4.14)$$

$$potential_i(R) = w_i \sum_{j \notin visited(R)} \frac{h_{ji}}{\sum_{k=0}^{n-1} h_{ki}} \quad (4.15)$$

$$w_i = (max_j h_{ji}) \cdot (1 - 0.1(\frac{c_{i0}}{max_j c_{j0}} - 0.5)) \quad (4.16)$$

**Lemma 1:** The time complexity of the proposed EERPS-DPDP scheme is  $O(B\mathcal{N}^2)$ .

**Proof:** The proposed EERPS-DPDP approach consists of two algorithms. The first is the EERPS algorithm for CH and RP selection, and the second is the DPDP-based optimal path selection algorithm for MS. In the proposed EERPS algorithm, the adjacency list is created first. Next, the nodes in the adjacency list are sorted in decreasing order of their capacity. The worst-case time complexity of the adjacency list creation algorithm is  $O(\mathcal{N}^2)$ . Furthermore, the worst-case time complexity of the sorting algorithm is  $O(\mathcal{N} \log \mathcal{N})$ . The time complexity for selecting CHs from the list of po-

tential CHs is  $O(\rho^2)$ . Therefore, the total time complexity of the EERPS algorithm is  $O(\mathcal{N}^2) + O(\mathcal{N} \log \mathcal{N}) + O(\rho^2) \approx O(\mathcal{N}^2)$ , where  $\mathcal{N}$  is the number of SNs in the networks. In the proposed DPDP-based optimal path selection algorithm, the heatmap is generated using GNN. Its complexity is  $O(p^2)$ , where  $p$  is the number of RPs. Next, the heatmap is updated for each solution and each feasible action. The complexity of this step is  $O(Bp)$ , where  $B$  is the number of solutions and  $p$  is the number of feasible actions. Parameter  $p$  is equal to the number of RPs. This step is repeated for  $p$  times, which makes the time complexity  $O(Bp^2)$ . Total time complexity is  $O(p^2) + O(Bp^2)$ . In the worst-case scenario,  $p = \mathcal{N}$ . Hence, the total time complexity of the proposed DPDP-based optimal path design algorithm for MS is  $O(\mathcal{N}^2) + O(B\mathcal{N}^2) \approx O(B\mathcal{N}^2)$ . Therefore, the total time complexity of the proposed EERPS-DPDP scheme is  $O(\mathcal{N}^2) + O(B\mathcal{N}^2) \approx O(B\mathcal{N}^2)$ .

### 4.3.3 MS-based data collection mechanism

After identifying the optimal MS traversing path, MS starts moving with constant speed to visit the first RP in the identified path. After reaching the RP, the MS sends a “HELLO” message. The “HELLO” message contains the CH ID, RP ID, and RP location. After receiving the message, CH checks the CH ID and RP ID. If IDs match, then CH transmits the data to the MS. MS receives data from the current CH and then continues to move to the next RP location. MS ends its data collection tour when it reaches the BS. Once MS reaches the BS, it uploads all collected data packets to the BS and again starts the new data collection cycle. In the EERPS-DPDP scheme, MS reaches all CHs individually through their nearest RP and collects data from them separately. It helps to gather data from each portion of the network.

**Lemma 2:** The message complexity of MS-based data collection mechanism is  $O(p)$ .

**Proof:** In the MS-based data collection mechanism, MS visits  $p$  number of CHs and sends  $p$  number of HELLO messages. Similarly,  $p$  number of CHs send their data packet

to MS. Thus, the total message complexity of the proposed MS-based data collection scheme is  $O(p + p) \approx O(p)$ .

**Lemma 3:** The total energy consumption by the network is  $\sum_{j=1}^p \left( \sum_{i=1}^{\mathcal{N}/p} \mathcal{E}_s^{ij} + \mathcal{E}_o^j \right)$ .

**Proof:** In the proposed EERPS-DPDP approach,  $\mathcal{N}$  number of SNs are grouped into  $p$  number of clusters. Each cluster contains average  $\mathcal{N}/p$  member nodes. The energy consumed by a cluster member SN for transmitting  $\mathcal{B}$  bits data to its CH is  $\mathcal{E}_s^{ij} = \mathcal{B}\mathcal{E}_c + \mathcal{B}\epsilon_{efs}d_{toCH}^2$ . The energy consumed in receiving  $\mathcal{B}$  bits data is  $\mathcal{E}_{rx}(\mathcal{B}) = \mathcal{B}\mathcal{E}_c$ . The energy consumed by CHs for receiving and aggregating data packets is  $\mathcal{E}_r^j = \mathcal{B}\mathcal{E}_{rx}(\mathcal{B}) \left( \frac{\mathcal{N}}{p} - 1 \right) + \mathcal{B}\mathcal{E}_{agg} \left( \frac{\mathcal{N}}{p} \right)$ , where  $\mathcal{E}_{agg}$  is the energy used for data aggregation. The energy consumed by CH for transmitting  $\mathcal{B}$  data bits to MS is  $\mathcal{E}_{tx}^j = \mathcal{B} \left( \frac{\mathcal{N}}{p} \right) (\mathcal{E}_c + \mathcal{B}\epsilon_{efs}d_{toMS}^2)$ . Total energy consumption at CH is  $\mathcal{E}_o^j = \mathcal{E}_r^j + \mathcal{E}_{tx}^j$ . Thus, the total energy consumed by the network is  $\sum_{j=1}^p \left( \sum_{i=1}^{\mathcal{N}/p} \mathcal{E}_s^{ij} + \mathcal{E}_o^j \right)$ .

**Lemma 4:** The data collection delay of the proposed MS-based data collection mechanism is  $\sum_{i=1}^p \left( \delta(R_i, R_{(i \bmod p)+1}) / \mathcal{V} \right) + (T_t \times p)$ .

**Proof:** In the proposed EERPS-DPDP approach, the time taken by MS to travel from one RP to another is  $\delta(R_i, R_j) / \mathcal{V}$ , where  $\delta(R_i, R_j)$  is the distance between two RPs and  $\mathcal{V}$  m/sec is the speed of MS. In the EERPS-DPDP scheme, MS speed is constant. The total time spent by MS for collecting data from CHs is  $(T_t \times p)$ , where  $T_t$  is the time spent by MS at an RP to collect data and  $p$  is the total number of RPs. Thus, the total data collection delay of the proposed EERPS-DPDP scheme is  $\sum_{i=1}^p \left( \delta(R_i, R_{(i \bmod p)+1}) / \mathcal{V} \right) + (T_t \times p)$ .

## 4.4 Performance Evaluation

This section evaluates the performance of the proposed EERPS-DPDP based intelligent data routing scheme for IoT-enabled WSNs. The simulations have been performed on the NS3 simulator. The experiments are carried out on a hardware platform featuring an Intel Core i7-9700 CPU operating at a clock speed of 3.00 GHz and equipped with 16.00

**Table 4.4:** Simulation parameters

<i>Parameters</i>	<i>Values</i>
Deployment Area	$100 \times 100 [m^2]$ , $200 \times 200 [m^2]$
Number of SNs	100 - 450 [#]
Initial Energy	0.5 [J], 1.5 [J], 3 [J]
Speed of Mobile Sink	1 - 4 [m/s]
Sensing Interval of SNs ( $SZ$ )	5 [s]
$E_c$	50 [nJ/bit]
$\epsilon_{fs}$	10 [pJ/bit/ $m^2$ ]
$\epsilon_{mp}$	0.0013 [pJ/bit/ $m^4$ ]

GB of memory. In this simulation, two deployment scenarios are considered. Scenario one considers  $100 \times 100 [m^2]$  area and scenario two considers  $200 \times 200 [m^2]$  area. The number of sensor nodes varies between 100 to 450. The initial energies of SN vary between 0.5 [J] to 3 [J]. SNs generate one data packet in every five seconds. Table 4.4 presents the parameter values which are used for simulation. The simulation parameter values are selected based on comparison approaches such as [65] and [62]. The proposed EERPS-DPDP approach is compared with TEO-MCRP [65], IRDA [37], EDEDA [62], and GA-SMT [38] in terms of stability period, network lifetime, residual energy, data collection delay, and data lost due to buffer overflow. The network scalability of the selected approaches is measured by the number of nodes in the network.

#### 4.4.1 Stability period and network lifetime

The stability period of a WSN is defined as the number of rounds until the first SN dies in the network. A high network stability period is a major requirement to increase the overall performance of the WSN. It also indicates that the network works without any partition/segmentation. Fig. 4.4a shows the network stability period in different node densities for scenario one. Fig. 4.4b depicts the network stability period in different node densities for scenario two. The EERPS-DPDP scheme increases the network stability period by 56% than GA-SMT, 44% than TEO-MCRP, 38% than EDEDA, and 29% than IRDA. This is because the proposed EERPS algorithm selects the optimal number of CHs that helps to reduce the transmission energy loss of SNs. Furthermore, it also creates efficient clusters that minimize the energy consumption of SNs. A high

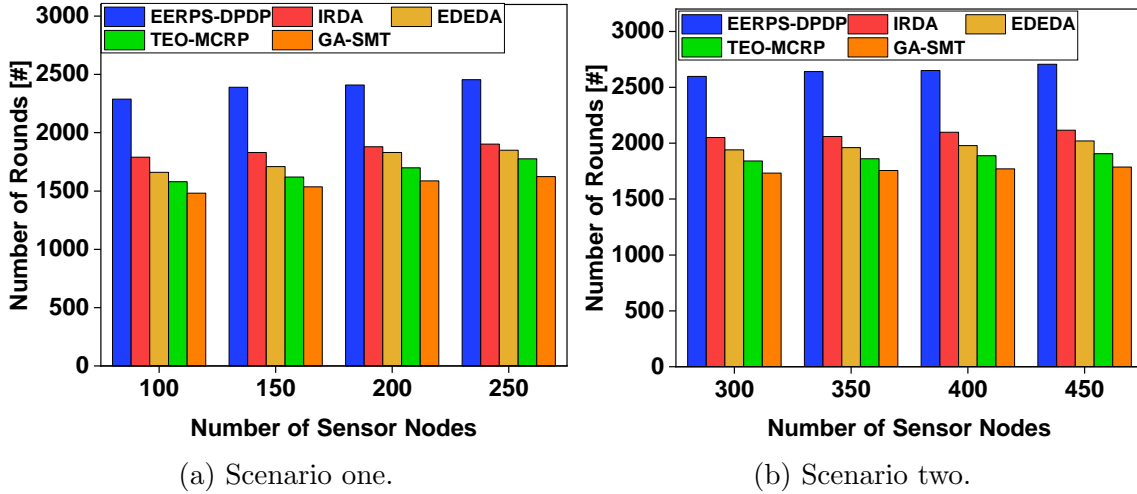


Figure 4.4: Stability period.

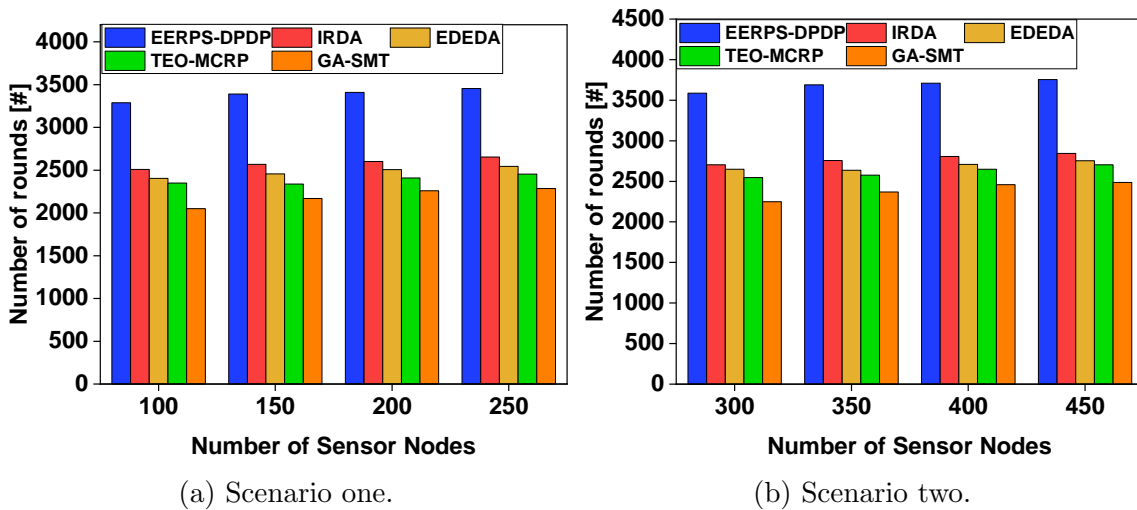
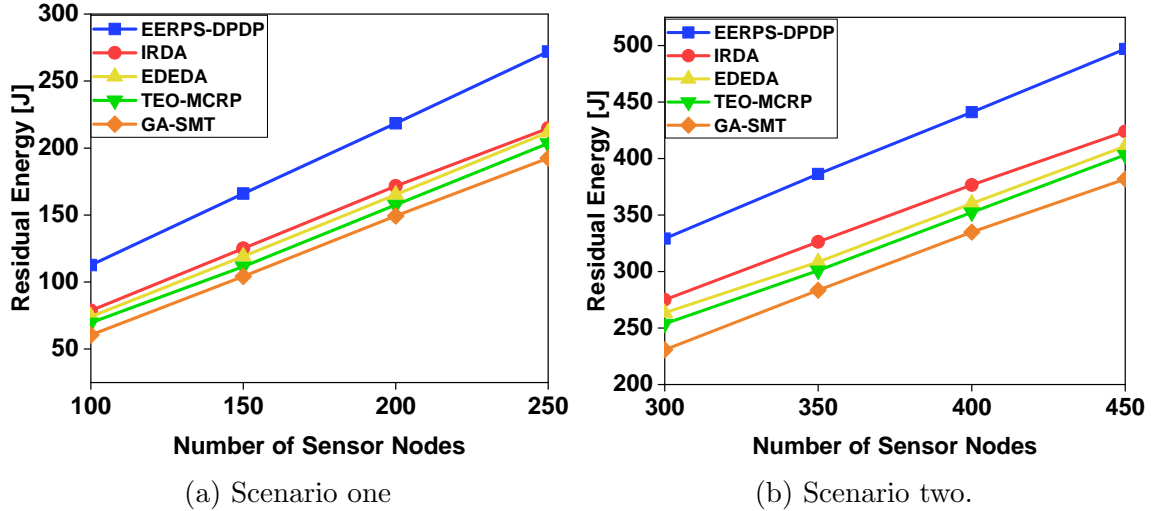


Figure 4.5: Network lifetime.

stability period also indicates that the proposed EERPS-DPDP scheme is more efficient for preventing premature death of the network. The network lifetime is the number of rounds until all of the nodes die in the network. High network lifetime shows that WSNs are able to operate for a longer period. Fig. 4.5a illustrates the network lifetime in different node densities for scenario one. Fig. 4.5b displays the network lifetime in different node densities for scenario two. The EERPS-DPDP scheme increases the network lifetime by 54% than GA-SMT, 42% than TEO-MCRP, 37% than EDEDA, and



**Figure 4.6:** Residual energy.

32% than IRDA. In the EERPS-DPDP scheme, the high-capacity nodes are selected as CHs. It reduces the load on low-capacity SNs and balances the energy consumption within the cluster. It saves energy at SNs and helps to increase the network lifetime. Similarly, the proposed MS-based data collection scheme also helps to reduce the data transmission distance of SNs. It significantly improves the network lifetime.

#### 4.4.2 Residual energy

Residual energy refers to the amount of energy remaining at SNs after performing some operations. High residual energy is a major requirement to increase the network lifetime. Fig. 4.6a depicts the total residual energy of the network in different node densities for scenario one. Fig. 4.6b displays the total residual energy of the network in different node densities for scenario two. The residual energy of a node is measured in Joules [J]. The EERPS-DPDP scheme increases the residual energy by 55% than GA-SMT, 46% than TEO-MCRP, 35% than EDEDA, and 31% than IRDA. This is because the EERPS-DPDP approach selects an optimal number of CHs that minimizes the transmission energy of SNs. Furthermore, MS reduces CHs' transmission energy loss by collecting data from the nearest RPs.

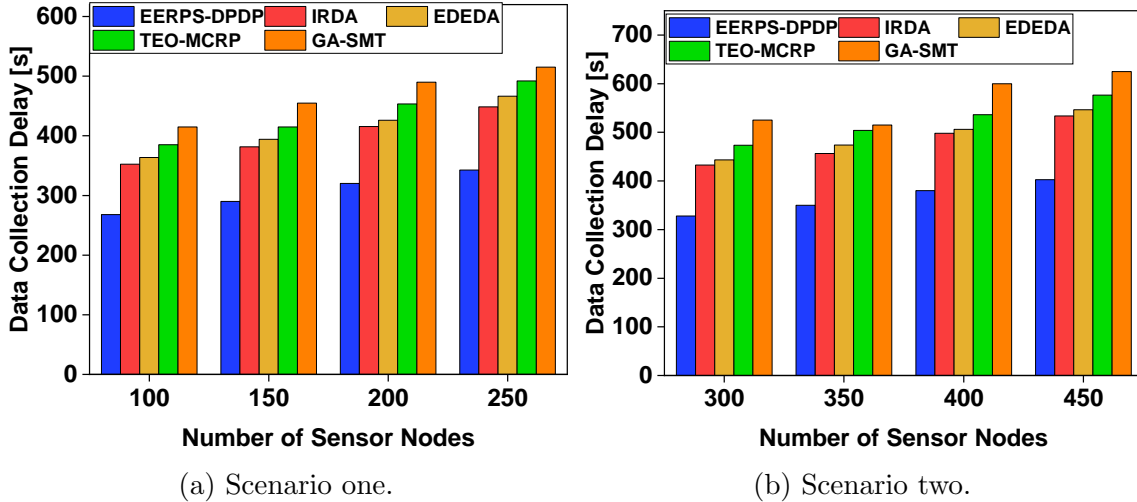


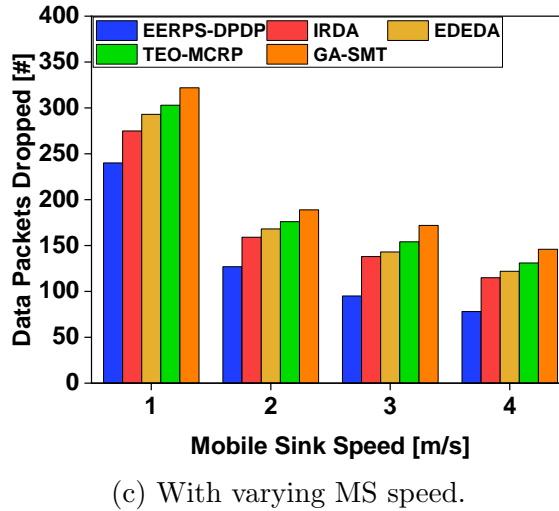
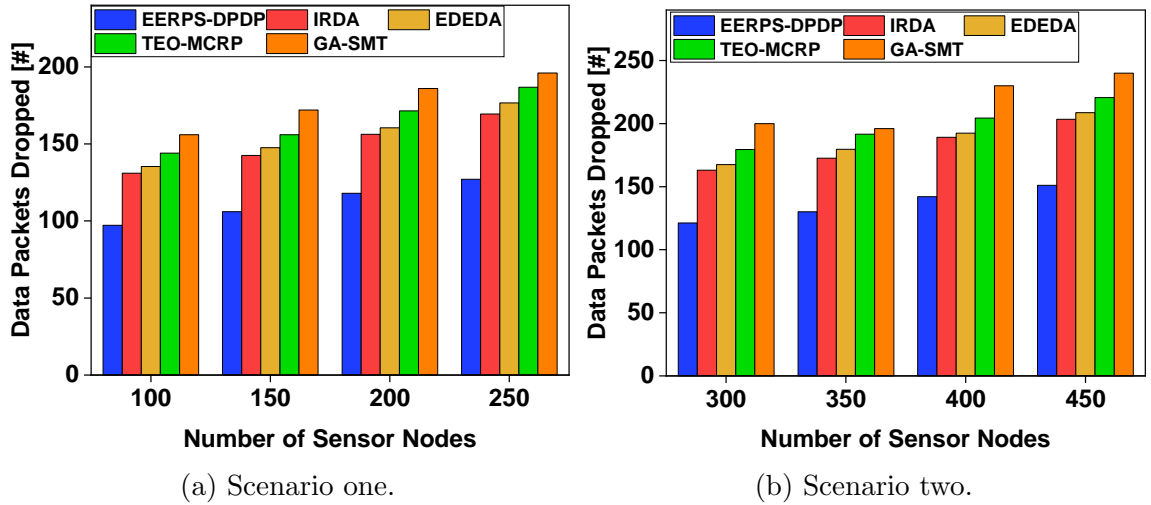
Figure 4.7: Data collection delay.

#### 4.4.3 Data collection delay

The data-gathering delay is the time difference between SN sensing the data and data reaching the BS. Data collection delay is measured in seconds [s]. In WSNs, low data collection delay is highly desirable to improve network performance. Fig. 4.7a shows the Data Collection Delay ( $DCD$ ) in different node densities for scenario one. Fig. 4.7b shows the  $DCD$  in different node densities for scenario two. The EERPS-DPDP scheme reduces the  $DCD$  by 53% than GA-SMT, 46% than TEO-MCRP, 38% than EDEDA, and 31% than IRDA. The proposed EERPS algorithm selects optimal number of RPs, and the DPDP-based algorithm designs the optimal path. These jointly help to reduce the data collection delay.

#### 4.4.4 Data lost due to buffer overflow

Fig. 4.8a depicts the Data Lost due to the SNs Buffer Overflow ( $DLBO$ ) in different node densities for scenario one. Fig. 4.8b shows the  $DLBO$  in different node densities for scenario two.  $DLBO$  is measured in terms of number of packets dropped. SNs have limited buffer space. If the buffer is full of previously stored data, then new data is



**Figure 4.8:** Data lost due to buffer overflow.

discarded by SNs. It leads to the loss of data. In a resource-constrained WSN, low data loss is very important to improve the performance and lifetime of the network. The EERPS-DPDP approach decreases data loss by 54% than GA-SMT, 47% than TEO-MCRP, 39% than EDEDA, and 34% than IRDA. Fig. 4.8c shows the  $\mathcal{DLBO}$  in different speeds of MS. The low speed of MS causes high data loss and long delays/bufferbloat problems. The data loss of the proposed approach is decreased by 53% than GA-SMT, 42% than TEO-MCRP, 35% than EDEDA, and 32% than IRDA. This is because the proposed EERPS algorithm selects CHs by considering the buffer capacities of SNs. It

helps reduce data loss due to buffer overflow. Furthermore, MS effectively collects data through an optimal data routing path that minimizes the data collection delay. It also helps to decrease data loss and prevent bufferbloat problems in the network.

The simulation results indicate that the proposed EERPS-DPDP scheme outperforms the existing approaches. This is because the existing EDEDA approach uses a grid construction technique in the network. This technique leads to high energy consumption at SNs and cell head nodes. It is also unable to optimize the data-gathering path, which causes poor network performance. GA-SMT and TEO-MCRP select CH every round, which leads to high message overhead and poor network lifetime. On the other hand, the IRDA approach performs continuous MS location prediction that leads to high message overhead and high energy consumption at SNs. The existing state-of-the-art approaches are unable to prevent premature death of nodes within the networks. It causes network segmentation. Furthermore, the existing state-of-the-art approaches do not consider the SN buffer capacity during CH selection. Therefore, they suffer from the huge data loss. The proposed EERPS scheme forms effective clusters and identifies optimal CHs in the network. It takes into account the SN buffer capacity, distance, and energy during CH selection. It reduces the network energy consumption and improves network lifetime and performance. Furthermore, the proposed scheme applies a DPDP-based optimal path design mechanism for MS, which reduces the delay and data loss due to the buffer overflow in the network. It effectively prevents premature death of the nodes and segmentation of the network.

**Table 4.5:** Comparison of complexities

<b><i>Approach</i></b>	<b><i>Time Complexity</i></b>
EERPS-DPDP	$O(B\mathcal{N}^2)$
IRDA	$O(P_s T \mathcal{N}) + O(\mathcal{N}^2 E \log \mathcal{N})$
TEO-MCRP	$O(\mathcal{N}^2 \log \mathcal{N})$
EDEDA	$O(\mathcal{N}^2 2^{\mathcal{N}})$
GA-SMT	$O(\mathcal{N}^3)$

Table 4.5 shows the comparison of time complexities. IRDA approach selects CHs with  $O(P_s T \mathcal{N})$  time where  $P_s$  is the population size,  $T$  is the number of iterations, and  $\mathcal{N}$  is the number of nodes. Furthermore, inter-cluster data transmission routes are computed with  $O(\mathcal{N}^2 E \log \mathcal{N})$  time complexity where  $E$  is the set of edges. Therefore, the total time complexity of the IRDA approach is  $O(P_s T \mathcal{N}) + O(\mathcal{N}^2 E \log \mathcal{N})$ . The EDEDA approach applies the Hamiltonian cycle to design the MS path based on grid cells. It requires  $O(\mathcal{N}^2 2^{\mathcal{N}})$  time. In the GA-SMT approach, the time complexity of the CH selection algorithm is  $O(\mathcal{N}^2)$ . Furthermore, it requires  $O(\mathcal{N}^3)$  time to calculate the MS path. Therefore, the total time complexity of the GA-SMT approach is  $O(\mathcal{N}^2) + O(\mathcal{N}^3) \approx O(\mathcal{N}^3)$ . In TEO-MCRP, the objective function for CH selection is computed with  $O(\mathcal{N})$  time. Additionally, nodes are sorted and divided into groups based on objective function value that requires  $O(\mathcal{N} \log \mathcal{N})$  time. Thus the total time complexity for CH selection is  $O(\mathcal{T} \times \mathcal{N} \times \mathcal{N} \log \mathcal{N})$ , where  $\mathcal{T}$  is number of iterations. After CH selection, the TEO-MCRP approach requires  $O(\mathcal{N})$  time to compute the MS movement path. Therefore, the total time complexity of the TEO-MCRP approach is  $O(\mathcal{T} \mathcal{N}^2 \log \mathcal{N}) + O(\mathcal{N}) \approx O(\mathcal{T} \mathcal{N}^2 \log \mathcal{N})$ . On the other hand, the total time complexity of the proposed EERPS-DPDP approach is  $O(B \mathcal{N}^2)$ , which is lesser as compared to the existing approaches. Fig. 4.9a shows the runtime in different node densities for scenario one. Fig. 4.9b shows the runtime in different node densities for scenario two. The proposed approach decreases the runtime by 55% than EDEDA, 48% than GA-SMT, 37% than TEO-MCRP, and 32% than IRDA. This is because the proposed EERPS algorithm forms optimal clusters with minimum time and message exchange. Furthermore, the DPDP-based optimal MS path design algorithm also significantly reduces the overall runtime of the proposed EERPS-DPDP approach.

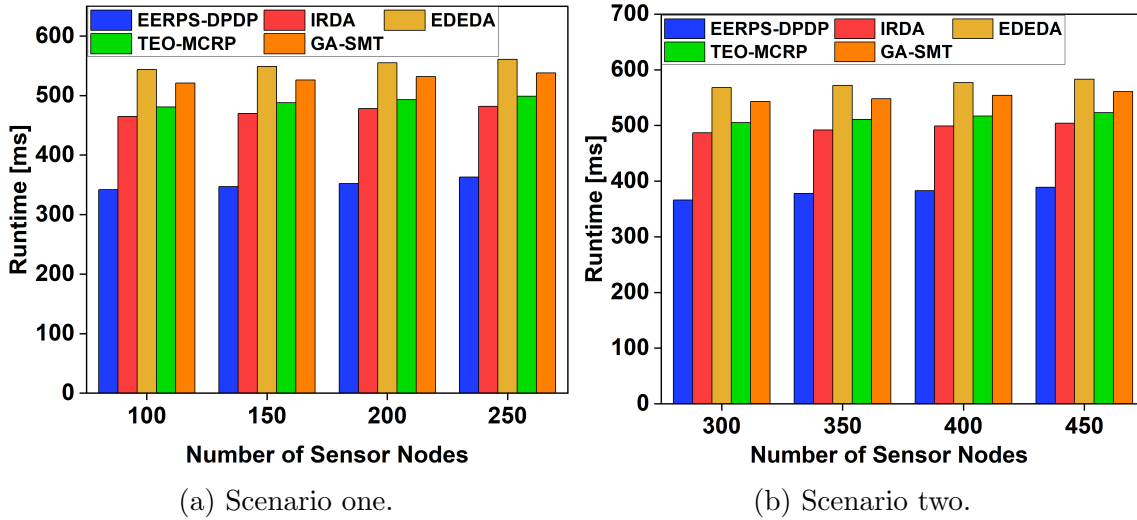


Figure 4.9: Runtime.

#### 4.4.5 Testbed experiment

This section analyzes the performance of the proposed EERPS-DPDP scheme in a real outdoor environment. Ten heterogeneous SNs are deployed in a  $20 \times 20$  [ $m^2$ ] area. Fig. 4.10 shows the SN deployment for the testbed. The SNs are composed of Arduino UNO with an XBee module for data transmission. SNs use IEEE 802.15 Zigbee protocol for data transmission. The SNs are equipped with temperature, humidity, and pressure sensors, which collect data from the monitoring environment. A robotic car is used as an MS. The experiment is conducted between 9:00 AM to 6:00 PM in an outdoor environment. The readings are collected every 30 minutes.

Fig. 4.11a shows the stability period and network lifetime. It is measured in minutes. The EERPS-DPDP scheme enhances the stability period up to 53% than GA-SMT, up to 42% than TEO-MCRP, 35% than EDEDA, and 27% than IRDA approaches. Similar to the simulation results, the EERPS-DPDP scheme shows a better stability period in the real testbed as compared to the other existing approaches. The EERPS-DPDP scheme also increases the network lifetime up to 51% than GA-SMT, up to 40% than TEO-MCRP, 34% than EDEDA, and 30% than IRDA. The EERPS-DPDP scheme

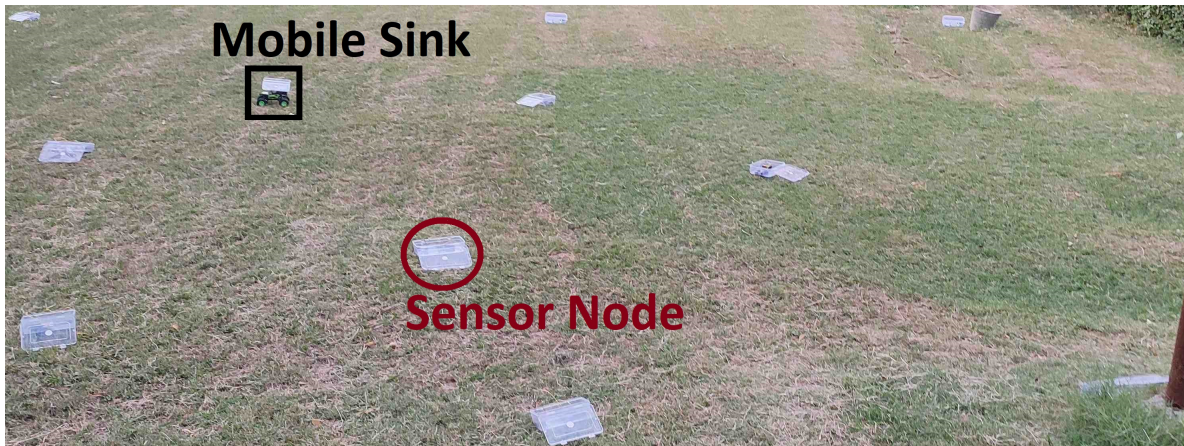


Figure 4.10: Sensor node deployment in the testbed.

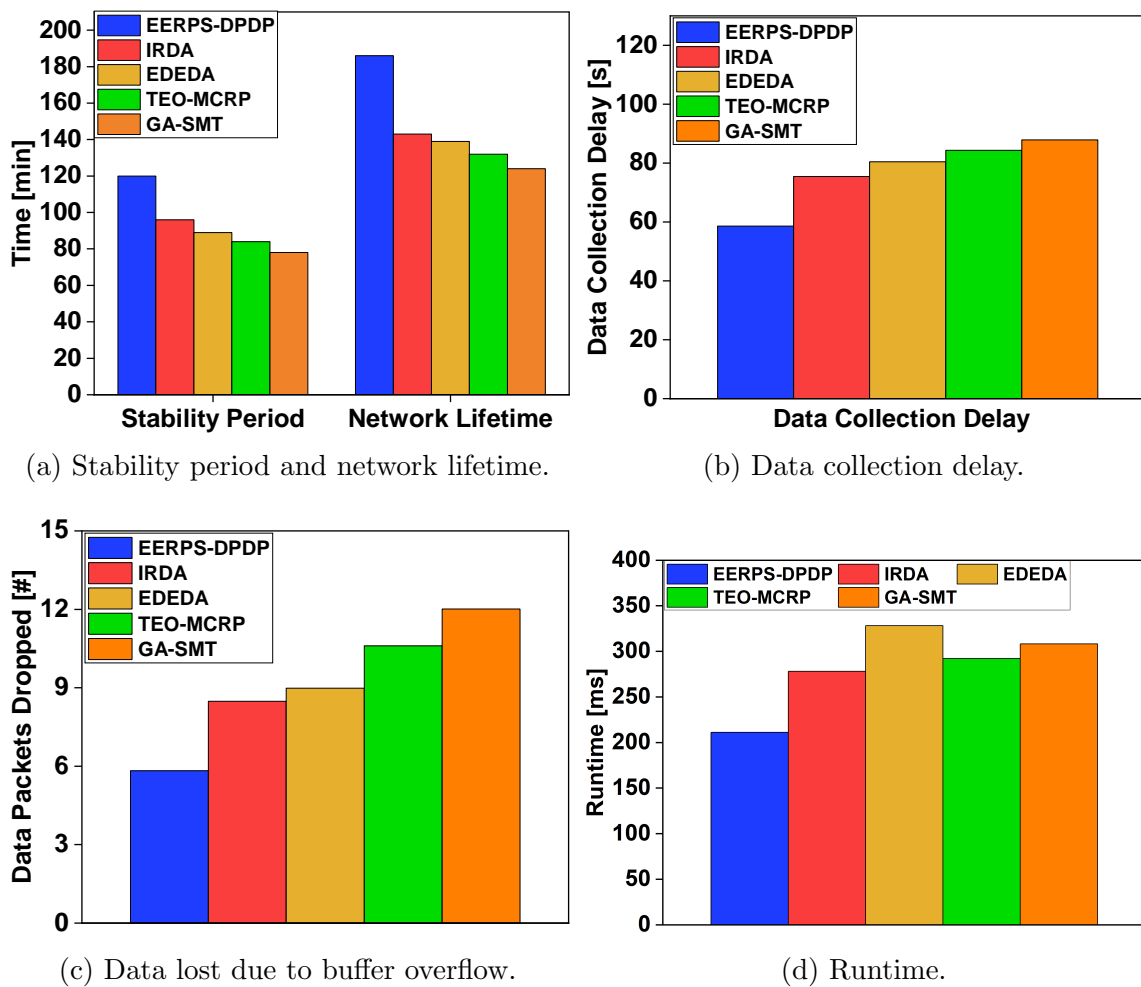


Figure 4.11: Testbed results.

shows a better network lifetime as compared to the existing algorithms similar to the simulation results. The proposed scheme selects optimal CHs that minimize the energy consumed in data transmission at SNs. It saves energy at SNs and increases the network lifetime and stability period. Table 4.6 shows the comparison of average residual energy. It is measured in Volts [V]. The result shows that the EERPS-DPDP approach improves the residual energy by 52% than GA-SMT, 43% than TEO-MCRP, 32% than EDEDA, and 28% than IRDA. Similar to simulation results, the proposed EERPS-DPDP scheme shows higher residual energy than other approaches. This is because the proposed EERPS-DPDP approach forms energy-efficient clusters and reduces the hops between SNs and CH. Fig. 4.11b shows the data collection delay in a real-life environment. Similar to the simulation results, the EERPS-DPDP scheme reduces the data collection delay up to 51% than GA-SMT, up to 45% than TEO-MCRP, 37% than EDEDA, and 29% than IRDA. The proposed DPDP-based path design scheme minimizes the distance travelled by MS, which reduces the data collection delay. Fig. 4.11c shows the data lost due to buffer overflow. The result shows that the EERPS-DPDP scheme reduces the data loss up to 51% compared to GA-SMT, 44% compared to TEO-MCRP, 36% compared to EDEDA, and 31% than IRDA. The EERPS-DPDP scheme shows a better result in terms of data loss, similar to the simulation results. This is due to the optimal data collection path design algorithm, where MS collects data from the CHs through the shortest data collection path. The difference between the simulation and testbed results is 2.79% due to the implementation error and other environmental conditions. Fig. 4.11d shows the comparison of runtime in the testbed environment. The proposed EERPS-DPDP approach decreases runtime by 51% than EDEDA, 45% than GA-SMT, 34% than TEO-MCRP, and 30% than IRDA. The proposed EERPS-DPDP approach shows the lowest runtime, similar to the simulation results. This is because the proposed EERPS-DPDP approach has the lowest time complexity as compared to existing approaches.

**Table 4.6:** Testbed results: average residual energy

<i>Time [min]</i>	<i>Average Residual Voltage [V]</i>				
	<i>EERPS-DPDP</i>	<i>IRDA</i>	<i>EDEDA</i>	<i>TEO-MCRP</i>	<i>GA-SMT</i>
30	11.18	10.55	10.35	10.21	10.09
60	10.34	8.9	8.73	8.28	8.24
90	9.48	7.32	7.11	6.57	6.31
120	8.55	5.55	5.31	4.72	4.12
150	7.67	4.46	4.22	3.26	2.18

## 4.5 Summary

In an IoT-enabled heterogeneous WSN, sensor nodes have different energy, sensing, communication, and storage capacities. Therefore, this chapter proposes a novel EERPS algorithm for IoT-enabled heterogeneous WSNs. It identifies optimal CHs for a heterogeneous network. It also selects optimal RPs for MS-based data collection from deployed heterogeneous sensor nodes. Furthermore, this work uses the DPDP algorithm to design an optimal data-gathering path for MS. Extensive simulation and testbed experimentation have been conducted to check the performance of the proposed scheme. The simulation and testbed results show that the proposed scheme performs better as compared to the state-of-the-art approaches.

Real life environment contains several large size obstacles Which prevent MS mobility within the network. It significantly hampers data collection and increases data collection delay. On the other hand collision between MS and obstacle can damage the MS physically. Therefore, an efficient obstacle avoidance scheme is very helpful for network performance improvement and in preventing collision between MS and obstacles. Next chapter proposes an MS-based optimal obstacle aware energy-efficient data routing scheme for IoT-enabled WSNs.