Chapter 4

NEW ALGORITHMS FOR SENSOR NODE DEPLOYMENT AND INTER-NODE COMMUNICATION

4.1 Introduction

The demand for oil is growing steadily from emerging and developing economies while oil field discoveries continue to decline. Therefore, the gap between demand and supply will increase with time. Subsurface Exploration deals with extracting valuable hydrocarbons from oil wells. Due to its hazardous nature, it’s one of the most difficult fields to carry experimentations on. Uncertainties associated with this field are result of various factors such as lack of information regarding location, size and spread of natural resource. In order to handle above listed factors, mobile wireless sensors seems to be a promising paradigm for increasing productivity and throughput by serving as intelligent investigators. At the time of this listing, none of the researchers has proposed the deployment of mobile wireless sensors in the oil fields. This work contributes a unique strategy that could deploy mobile sensors in the subsurface to get real time information which otherwise is not possible.

Chapter 3 not only dwelled upon various challenges of subsurface exploration but also explored the graved potential of WSN in the domain under consideration.

Primary technological challenges are

- Percentage of hydrocarbons recovered from wells is suboptimal. Current recovery status is 30-35% and an obvious demand is to increase this percentage.
- Maintaining and monitoring pipeline integrity (e.g., cracks, leaks)
- Deployment of sensors into the wells and retrieving the information.
• Establishing communication among deployed collection of sensors.
• Processing large amounts of raw data for human visualization
• Maintaining the topology as well as connectivity of deployed sensors

Based on the investigated limitations and latest requirements presented by AEC [9], following objectives are being identified:

• To propose a novel and efficient deployment strategy of sensors in the subsurface area.
• To propose a communication technique to establish an effective inter-networking within the deployed collection of nodes.
• To propose an efficient strategy for information processing using collaborative signal processing and filtering of noise from the sensed data volumes.
• Introducing fault-tolerance and robustness in the proposed model by proposing a node replenishment strategy.
• Evaluation & comparison of proposed work with existing algorithms proposed for sensor application on land.

In order to achieve the above stated objectives, the work preceded in four phases wherein evaluation of the proposed model is being carried out at the end of each phase. Next section presents the flow of work to make the floor smooth for oil extraction companies and scientists intending to use the sensed data for real-time interpretations.

4.2 High Level View of the Proposed Work

The work started with the primary objective of meeting the stated technological challenges. As shown in figure 4.1, the four phases namely, Deployment Phase, Communication Phase, Information Processing Phase and Maintaining Robustness of the proposed work is briefly discussed as follows:
(A) **Sensor Node Deployment Phase**

First phase deals with the major challenge of deploying sensors in the oil field, which, practically is unreachable. So in order to deploy sensors, some remote device is essential. This work owes credit to the pipeline robot developed by Iyengar and his team in the Robotics Research Lab [116] at Louisiana State University for initially deploying the sensor in area just near well-head. The phase proposes a novel deployment strategy meeting the optimal coverage-connectivity and ease of deployment challenges presented in chapter 2 i.e. Minimal number of sensors could be deployed easily ensuring the connectivity among all. The detail of deployment algorithm is being presented in the upcoming section 4.3.

(B) **Inter-node Communication Phase**

After deployment, a strong need for communication is apparent. This phase proposes a query driven routing strategy that aims to in-correlate diverse sensor input generated beneath the earth. The proposed communication strategy
behaves as a multistage data collection system that clubs the data from various sensors and the same can be used to detect and track comprehensive events that go unnoticed using conventional drilling and mining tools. The proposed algorithm is being discussed in the section 4.4.

(C) Information Processing Phase

The data sensed and hence forwarded to the base station is usually corrupted with noise. In order to remove the noise, filtering is desirable that otherwise would shorten the lifespan of sensors. This contribution presents a unique application of Kalman filtering technique for processing such sensitive information because sensor readings are usually imprecise due to strong variations in environment and also, computation has to be much more energy efficient than communication. Chapter 5 discusses this phase in detail.

(D) Maintaining Robustness

While implementing the third phase, it was discovered that sensors are prone to drift and it is important to perform post job calibrations to confirm the accuracy of the data. The application requires high degree of reliability mechanism in to reduce end-to-end packet loss ratio. Drifting nodes and failing nodes due to depletion of battery power or other reasons raise a significant number of routing issues. Therefore, a robust node replenishment algorithm is proposed that replenishes the void created due to a failed or drifted away node and the details of the same are given in Chapter 5.

A diagrammatic view of the proposed architecture beneath the earth is given in Figure 4.2.
Upcoming sections describe the proposed phases in detail. It shall be noted that each phase has been evaluated individually and hence the evaluation has been given in the respective section only.

4.3 The Sensor Node Deployment Phase

As already stated and shown in figure 4.2, this phase owes the credit to the agent robot developed Iyengar & Dimple at Robotics Research Laboratory, LSU. The robot used for deployment of sensors is a pipeline robot, which not only can travel in a straight pipe but also can traverse an L-shaped pipe. Figure 4.3(a) and 4.3(b) shows the prototype of pipeline robot and its vertical mobility respectively.

The proposed deployment algorithm adopts an incremental deployment strategy that makes use of information provided by previously deployed nodes and is inspired from the works of Andrew et al. [5] who made an attempt to systematize non deterministic deployment, by using the information gathered from previous phase. However, this work proposes A Novel and Efficient Algorithm for Deploying Mobile
Sensors in Subsurface [54] considering the harsh conditions and real-time environment prevailing in subsurface.

Figure 4.3 (a): Prototype of Pipeline Robot
[Photo courtesy RRL, LSU, USA]

For instance, in contrast to the author’s presumption of homogeneous nodes, this work focuses on heterogeneous nodes. The proposed algorithm aims to maximize both the network coverage as well as the connectivity parameter. Before presenting the algorithm in detail, following are the key considerations that are to be assumed as is for convenience and simplicity.
4.3.1 Key Assumptions

The proposed algorithm has been developed keeping the following three factors as its key assumptions.

(A) Heterogeneous nodes

The objective is to dynamically guide the sensors to its final location. The increased stability and appropriate location of the sensors helps reducing the number of measurement points required to monitor precisely a surrounding area and consequently reduce the cost of a survey. In order to measure the modalities present within subsurface, various kind of sensor nodes such as temperature sensor, pressure nodes, acoustics nodes, flow nodes etc. will be deployed.

(B) Geometrical Considerations

It is assumed that the gross topology within the subsurface remains static while the network is being deployed. This implies that the geo obstacles
present do not displace during this course of time. One of the major design challenges is simplified i.e. the sensor network be reconfigured, expanded, or otherwise modified without a loss in performance. The static configurations and topologies are conducive to scalability in terms of power and bandwidth usage, congestion control, and overall network robustness to disruptions.

(C) Mesh Communication

Assuming that each sensor has a sensing radius, “r” and it can only sense the environment and detect events within its sensing area, which is the disk of radius “r” centered at the sensor. A point is assumed to be covered by a sensor if it is located in the sensing area of the sensor. The sensor network is thus partitioned into two regions, the covered region, which is the region covered by at least one sensor, and the uncovered region, which is the complement of the covered region. A communication is said to be mesh or full if it detects the events happening in both covered and uncovered area. In addition, it is assumed that all nodes after being deployed in the network can communicate with remote base-station.

Based on these three assumptions, following sub-section presents a high-level view of the proposed algorithm.

4.3.2 High Level View

The algorithm primarily comprises four phases namely, Initialize, Choose, Allocate and Enforce. The flow of information among various phases is as depicted in figure 4.4 and the key functionality of the phases is delineated in Table 4.1. It may be noted that the nodes are in either waiting, active or deployed state and conditions for each state change is explained with the help of figure 4.5.
The algorithm iterates through the Choose, Allocate and Enforce phases, terminating only when all nodes have been deployed. The phases are explained in the following subsections.

Table 4.1: Phases and their Key Functionalities

<table>
<thead>
<tr>
<th>Phases</th>
<th>Key Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>Set the state of a node to waiting, active or deployed.</td>
</tr>
<tr>
<td>Choose</td>
<td>Develop a location map with the help of deployed nodes and select the appropriate location for waiting node.</td>
</tr>
<tr>
<td>Allocate</td>
<td>Assign destination to a waiting node and set the state as active.</td>
</tr>
<tr>
<td>Enforce</td>
<td>Active node is deployed at the desired location successively</td>
</tr>
</tbody>
</table>
(A) Initialize

Initialize is the very first phase of the algorithm and initially all nodes are set to waiting state except for one whose state is set to active and is considered as a node ready to be deployed.

(B) Choose

This phase determines the next deployment location, or goal and the desired goal is usually the one that either maximizes the coverage metric i.e. $\text{max(coverage\_metric)}$ where coverage metric is computed at a location ‘l’ for a modality ‘m’ by maintaining an allocation grid. An Allocation Grid is 2D arrangement of the total area that is likely to be covered. Each cell of the grid is defined as unallocated, allocated or unknown. An unallocated cell implies that it contains no obstacles whereas an allocated cell state implies that it can pose an obstacle. Here obstacle implies a node, which has been deployed in that particular cell. It is being termed as obstacle since it might obstruct the deployment of other nodes to farther cells. In case of non-availability of information about the state of a cell, it is treated as an unknown location i.e. the unknown location cannot be chosen as the next desired destination.

Now, each deployed node communicates its sensing range say ‘r’ implying that the upcoming active node shall be deployed in a cell which is not only set as unallocated but also satisfies the relation $c \geq 2r$, where ‘c’ is the communication range ensuring the condition $\text{max(coverage\_metric)}$. Depending upon the information gathered, the appropriate cell is chosen and the location of the same is forwarded to the next phase.

(C) Allocate

The main aim of the phase is to assign the location chosen during chose phase to a waiting node.
The state of the node changes from waiting to active. The phase allocation is simple in case the desired location can be reached unobtrusively. However, if there are nodes already deployed on the path to be followed by the active node, these deployed nodes act as obstruction and such obstructions become increasingly likely to happen as the size of the network increases. This makes “Allocate” the most important and complex phase of deployment. Now, in order to overcome the stated complexity this work proposes a novel
mechanism, which allows goal swapping i.e. an active node swaps its own goal with the deployed node leading to the state exchange between two nodes. This implies that the active node acquires the location of deployed node and the deployed node becomes the new active node.

(D) Enforce

During this phase, active nodes are ultimately deployed to their goal locations. Node deployment is a consecutive process i.e. only one node remains in active state while all other are either waiting or deployed. Since there is only one node in motion at any given point in time and the goal assignment ensures that each successive goal is unobstructed, there is very little possibility of interference between nodes. However, this deployment process is quite slow. The execution time is proportional to the sum of the distances traveled by the active nodes which in turn is equal to the distance a single node would have traveled if there were no obstructions. As the area covered by the deployed network becomes larger, nodes will have to travel farther and farther and hence it is expected that execution time will increase with the number of deployed nodes.

The upcoming section presents the working algorithm, flowchart of the proposed work, followed by an example of node replacement.

4.3.3 Working Algorithm, Flowchart & Mathematical Model

The section is divided into four parts i.e. proposed algorithm, flowchart, an example and a mathematical model proving that this proposed algorithm over shadows the other existing algorithms in allied fields. A comparison with the Kar and Banerjee [91] is being provided at the end of the section.
A) Algorithm for the proposed deployment strategy:

Input required for the algorithm:

- node(i): which represents the $i^{th}$ node to be deployed
- $g_1$: which represents the randomly selected goal where the node(1) is to be deployed.

1) initialize: set status [node(i)] = waiting $\forall i \in \{1,2,\ldots,m\}$
2) set status [node(1)] =active
3) deploy node(1) at goal $g_1$
   set status[node(1)] =deployed
4) repeat
5) collect information from node(i) $\forall i \in \{1,2,\ldots.m\}$
   calculate new goal $g_j \forall j \in \{2,3,\ldots.m\}$
6) if node(i) $\forall i \in \{1,2,\ldots.m\}$ lie on linear path between point of deployment and $g_j$ then
7) set status [node(i)] = active $\forall i \in \{1,2,\ldots,m\}$
   redeploy nodes
8) deploy node(i+1)
9) set status[node(i+1)], status[node(i)] = deployed
10) else
11) deploy node(i+1) to goal $g_j$
12) set status[node(i+1)] = deployed
13) end if
14) until status[node(i)] =deployed $\forall i \in \{1,2,\ldots,m\}$
15) stop
B) Flowchart:

Figure 4.6: Flowchart of Proposed Deployment Strategy
C) An Example

As already stated the proposed algorithm follows an iterative approach, hence, if node\((i)\) obstructs the path to node\((i+1)\) deployment location, then node\((i)\) can be redeployed to node\((i+1)\) location, while node\((i+1)\) will occupy the previous location of node\((i)\) as demonstrated in figure 4.7(a),(b) and (c) respectively. As can be seen, node 4 is the next to be deployed. However, node 1, 2 and 3 are obstructing the desired location. In this case, the proposed algorithm executes in order that node 3 increments to next available location creating a void for node 2 to occupy and node 1 in turn occupy the void created by node 2 thereby creating space for node 4.

![Figure 4.7: Iterative Deployment of Nodes](image)

Since all nodes are assumed to be equivalent, this goal-swapping makes no functional difference to the network. For complex environments, with many obstructions, this resolution strategy may need to be applied recursively, so that

\[
\text{node}\,(j-1) \sim \text{replaces} \sim \text{node}\,(j) \quad \forall \ j \in i - 1, \ i - 2, \ldots, \ 1
\]  

[Fig 4.7(b)]
here, status \( \{\text{node}(i)\} = \text{active} \) i.e. it’s being deployed in the current iteration. The proposed mechanism executes in an iterative manner oozing the nodes from a central location to an uncovered location. In addition, as the nodes spread throughout the environment, the same nodes will tend to remain on the edge of the network.

It is important to note that how much distance shall a node move i.e. next optimal location of deployment needs to be evaluated and this is the main crux of proposed strategy. Next part achieves this objective.

**D) Mathematical Model**

The proposed algorithm with the mentioned assumptions and constraints \((c \geq 2r)\) would lead a 2D allocation grid as shown in figure 4.8. The network is formed by placing several horizontal r-strips. The paper [91] defines r-strip as a string of r-disks placed along a line such that the distance between the centers of any two adjacent r-disks is \(r\). Sensors with a sensing/transmitting radius \(r\) can be modeled as a disk with radius \(r\). This disk is being referred as an r-disk.

![Figure 4.8: A 2D Allocation Grid](image-url)
These r-strips are positioned such that the x and y co-ordinates of each node are given by:

\[
\begin{align*}
  x_{ij} & = \begin{cases} 
    \pm \frac{ir}{2} & \text{if } j \text{ is odd} \\
    \pm \frac{(2i-1)r}{2} & \text{if } j \text{ is even}
  \end{cases} \\
  y_{ij} & = (j-1)\sqrt{15} \frac{r}{2}
\end{align*}
\]  
\[\text{…………(eq. 4.1)}\]
\[\text{…………(eq. 4.2)}\]

where \( i = 0, 1, 2, \ldots \) and \( j \) refers to the r-strip number = 1, 2, 3, ….

These r-strips are connected to each other by inserting some more sensors in the network, which will be referred as Global Connectivity Sensors (GCS). The positional co-ordinates of GCS are:

\[
\begin{align*}
  x_i & = (-1)^i \frac{r}{4} \\
  y_i & = (2i+1) \sqrt{15} \frac{r}{4}
\end{align*}
\]  
\[\text{…………(eq. 4.3)}\]
\[\text{…………(eq. 4.4)}\]

where \( i = 0, 1, 2, \ldots \)

Let us consider a portion of the network consisting of the sensors 1–6, as shown in figure 4.9.

![Figure 4.9: Coverage and Connectivity Description](image)
The dots (●) represent deployed sensors nodes. The sensors 1 and 3 are located within two odd-numbered r-strips, whereas, the sensors 2 and 4 are within an even-numbered r-strip. The GCS (sensors 5 and 6) are positioned to connect the sensor 1 to 4 and 2 to 3 and hence, the corresponding r-strips are connected to each other. Such a placement of GCS allows an increase in the vertical spacing between the r-strips as compared to the network described by Kar and Banerjee [91]. This eliminates any overlapping between the r-strips (See Fig. 4.9) and hence, an increase in the coverage area as compared to [91], the details of which are presented in the next subsection.

4.3.4 Performance Evaluation and Comparison

As per the paper [91], nodes of adjacent r-strips are inter-connected with each other by inserting new nodes in between them (Node E & F in this case). Here node A & C lies on odd r-strips, while node B & D lies on even r-strips (see figure 4.10). To ensure connectivity, maximum permissible distance between two deployed nodes (B & D) = r. Sensor E & F had to be so placed that their distance from A, B, C, D = r.

![Figure 4.10: Detailed view of r-strips placement as per [91]](image-url)
Since distance between node B and node D is equal to r, nodes E & F are placed parallel to nodes A & C. Therefore

\[ BO=OD=r/2 \]  \hspace{1cm} (eq. 4.5)

Now let's compute EO from right angled triangle BOE,

\[ \begin{align*}
BE^2 &= BO^2 + EO^2 \\
r^2 &= (r/2)^2 + EO^2 \\
EO &= \sqrt{(r^2-r^2/4)} = \sqrt{3r}/2
\end{align*} \]  \hspace{1cm} (eq. 4.6, 4.7, 4.8)

This gives,

\[ \begin{align*}
AO &= AE + EO \\
&= r + \sqrt{3}r/2 \\
&= 1.866r
\end{align*} \]  \hspace{1cm} (eq. 4.9, 4.10, 4.11)

which, basically is the vertical spacing between two r-strips as proposed by [91]. Similarly, compute AB from right angled triangle AOB,

\[ \begin{align*}
AB &= \sqrt{BO^2 + AO^2} \\
&= \sqrt{(r/2)^2 + (r + (\sqrt{3}/2)r)^2} \\
&= 1.931r < 2r
\end{align*} \]  \hspace{1cm} (eq. 4.12, 4.13, 4.14)

Hence, there is overlapping between r-disks lying on alternate r-strips. This is the limitation of previous work. In order to overcome this limitation, our work proposed to increase vertical spacing so as to reduce overlapping. This can be done by inserting the interconnecting nodes E & F at points as specified in Fig. 4.11.
With such placement, we would have

\[ \text{BO=OD}=\frac{\pi}{2} \quad \ldots \ldots \text{ (eq. 4.15)} \]

and

\[ \text{AB}=2r \quad \ldots \ldots \text{ (eq. 4.16)} \]

Now let’s compute the vertical distance (AO) between alternate r-strips,

\[ \text{AO} = \sqrt{\text{AB}^2 - \text{BO}^2} \quad \ldots \ldots \text{ (eq. 4.17)} \]

\[ = \sqrt{(2r)^2 - \frac{r^2}{4}} \quad \ldots \ldots \text{ (eq. 4.18)} \]

\[ = \sqrt{15/2} \text{ r} = 1.936 \text{ r} \quad \ldots \ldots \text{ (eq. 4.19)} \]

The value of AO reflects that our algorithm has been able to increase the vertical spacing between two r-strips, which in turn would lead to reduced overlapping and the number of sensors to be deployed. It implies that overall time of deployment of sensors, the cost of deployment, and the cost of sensors is reduced.

Figure 4.11. Detailed View of r-strips Placement as per our Proposed Strategy
significantly. The result clearly indicates that our algorithm and placement pattern provides connected coverage to entire field.

4.4 Inter-node Communication Phase

This section proposes a query driven routing strategy that aims to in-correlate diverse sensor input generated beneath the earth. The proposed communication strategy behaves as a multistage data collection system that clubs the data from various sensors and the same can be used to detect and track comprehensive events that go unnoticed using conventional drilling and mining tools. The strategy is able to describe dynamic behavior of complicated subsurface and its exploration, allowing us to detect real time events and differentiate among various events happening simultaneously, that needs to be addressed on priority basis. The strategy is a unique contribution to subsurface exploration technology as none has proposed the deployment and hence communication strategy for this field, in particular.

Unlike traditional communication networks, application of sensors within subsurface requires to flow accumulated data based on various parameters to the base station. This could be done in single hop (as in Direct Reporting) or via multiple hops (as in Directed Diffusion). The other apparent factor to be considered is the resource constraints within wireless sensor nodes. These nodes have limited energy, processing and storage capabilities. Keeping in mind the resource constraints, we have proposed A Query Driven Routing Protocol for Wireless Sensor Nodes in Subsurface [53], since it requires transmission only on “as and when required” basis.

Next section present the design challenges for the proposed routing protocol.
4.4.1 Design Challenges of Routing Protocols

The first and foremost goal of any WSN is to prolong the lifetime of the network, while maintaining the communication quality. The design of routing protocols is dependent upon various issues which designers consider while zeroing upon an efficient communication technique. Some of the major issues to be considered are:

(A) Maintaining trait between precision and energy consumption

The energy source for sensors is always limited. The available energy is used both for communication as well as in-node data processing. Hence, energy management must always be taken care of while deciding upon a protocol.

(B) Node Deployment

The traditional deployment of sensor nodes can be either deterministic or non-deterministic. In case of deterministic deployment, data flow along a predetermined route. While in the case of non-deterministic deployment, the nodes are not placed uniformly, hence the communication is generally multi-hop. In our previous work we have laid down the deployment strategy for mobile nodes within subsurface, which led to a fixed location pattern of mobile nodes. These nodes have had an optimal vertical spacing; hence directed diffusion becomes the choice for data flow within them.

(C) Data Aggregation

Since in WSNs a huge amount of redundant data is being generated, several techniques like minima, maxima, average, duplicate suppression can be used for Data Aggregation. While choosing any routing protocol data aggregation always remain one of the most important factors, since communicating redundant data will lead to wastage of scarce and valuable energy sources.
(D) *Quality of Service*

In certain of the WSN applications, delivery of sensed data is highly time critical, since otherwise it is rendered useless. Examples of such applications are detection of hurricane or certain other calamities. While in other applications, time criticality is not involved, and the data may be transmitted as per eases. In such cases, special care can be taken to maximize the lifetime of network as well as available energy resources. Example of such applications can be habitat monitoring.

(E) *Fault Tolerance*

In any of the application, some nodes might fail due to unavoidable factors like energy, damage or environmental interference. In case of such a failure, the working of overall network should not be hampered and it should continue with its normal tasks. The routing protocol should be so robust as to create new links.

(F) *Scalability*

The routing protocol should be scalable enough to adjust any number of nodes within network. There can be just hundreds of nodes at time, while at other times there can be thousands of nodes to cater to.

(G) *Data Reporting Model*

Data reporting model of WSNs fluctuate significantly depending on the application and on the time criticality factor of the data. Data reporting can be time-driven, event driven, query-driven or even a hybrid model of these. The routing protocol is predominantly dependent of the choice of data reporting model.
**Connectivity**

Sensor nodes are never secluded from one another and are anticipated to be highly connected.

**Coverage**

A sensor node has to cover a certain physical range in its working environment. The range is normally limited. In our proposed mathematical model (explained in next section), its value had come out to be $1.936r$ [54].

### 4.4.2 The Proposed Routing Protocol

The proposed routing strategy derives its base from previous deployment phase. The resultant placement of nodes in subsurface is as shown in Figure 4.12. The filled dots represent mobile nodes deployed initially, forming r-strips, while the hollow circles represent GCS nodes deployed at later stage to interconnect these r-strips. The interconnecting zig-zag nodes (hollow ones) are placed near the base station or the oil well head, since otherwise the communication path would be too long for the nodes. Figure 4.13 depicts the directed diffusion used for this particular case.

In Directed Diffusion Communication Technique, each node reports to its nearest neighbor. This message is relayed by other nodes until it reaches the base station. This communication technique is superior to that of Direct Reporting since it does not create any bottlenecks, and even those nodes which are far away from the Base Station can pass on their message to the Base Station in a multi hop fashion.

The biggest limitation of this technique is that nodes near base station usually have a short life span. To overcome this problem, data aggregation techniques can be used. Directed Diffusion is a popular data aggregation paradigm for WSN routing.
This protocol, as proposed by C. Intanagonwiwat et al. [28], is a data-centric and application-aware protocol in the sense that all data generated by the sensor nodes is represented by attribute-value pairs.

Figure 4.12: Node placement within subsurface

The protocol aggregates data coming from different sources enroute, while eliminating redundancy, reducing the number of transmissions and hence saving on scarce network resources like energy and processing capabilities.

Figure 4.13: Directed Communication within nodes
Contrary to conventional end-to-end routing, Directed Diffusion [27] has the capability of discovering routes from multiple sources to a single destination, allowing in-network aggregation of superfluous data. In this protocol, sensors evaluate events and create gradients of information in their individual vicinity. The base station seeks data by sending interests.

Our work proposes to exploit Query Driven Data Reporting Models (QDRRM), so that the nodes do not wear off too soon. The interest generated by the routing protocol is used to describe a sensing task required to be done by the network. This interest diffuses through the network hop-by-hop and each node broadcasts it to its neighbors. As this interest passes hop by hop through the sensor nodes, it on way creates gradients, which in turn collects the data to satisfy the query as generated by the base station.

A) Incorporating Mobility into Directed Diffusion

As already illustrated Directed Diffusion can be used in case of subsurface exploration. However in order to enhance the efficiency of routing strategy, we propose to exploit mobility characteristics of sensors used while deployment. The deployment strategy [54] could form r-strips along with connecting nodes known as GCS (Global Connectivity Sensors). These GCS’s has the potential to move parallel to r-strips in order to collect data from sensors. Now since for subsurfaces, data reporting is not event driven, hence it can be collected by GCS as per convenience, aggregated and forwarded to next r-strip or base station.

It is assumed that during the course of communication, nodes would remain static. The deployed GCS act as collectors responsible for not only data collection but also to take intelligent routing decision. The proposed mechanism works in two phases: *Forward data collection phase* and *backward data collection phase*. The two phases and the working algorithm are given in the following subsections.
(i) **Forward Data Collection**

There are ‘a’ sensors numbered 0…., a−1, within a single r-strip, such that the collector follows the cycle of 0→1→...→a−1→a−2→…→1→0. When the GCS arrives at an r-strip sensor, it stays there long enough to collect the data accumulated since last visit of the collector, emptying the sensor’s buffer. Let \( \alpha_i \) be sensor data accumulation rate at individual sensor node \( i \), where \( i \in [0, n-1] \). Let \( \beta_i \) be the data collection rate of the mobile collector when its visits the \( i \)th node. We assume that \( \beta_i > \alpha_i \); otherwise, the sensed data will eventually be lost. The time to collect data from each head is divided into two parts: the time for travel to the head and the time for transferring the data. Suppose \( d_i \) is the time for collector to travel from \( i \)th sensor to the \((i+1)\)th sensor (modulo \( n \)). We use \( t_i \) to represent the time to collect data from the \( i \)th head \((t_i > 0)\). The total time ‘\( T'\) is the sum of time duration taken by GCS collector to visit all sensor nodes once, along with time taken for collecting the data.

\[
T = 2 (\sum_{i=0}^{a-1} (d_i + t_i)) \quad \text{......... (eq. 4.20)}
\]

The proposed routing algorithm assumes that mobile GCS collectors have intelligent routing decision capabilities. While moving parallel to r-strips, they collect data from sensors and also decide the appropriate sensor where this data is to be dropped. By exploiting the mobility characteristics of GCS, we can change interconnection between the nodes and thus obtain different network topologies. We model the routing protocol as a multiple hop transmission taking number of hops as our cost metric. This problem can be solved by any shortest-path routing algorithm. The shortest path is to be decided and stored as meta-information within all GCS. These r-strips are positioned such that the x and y co-ordinates of each node are given by:
\[ x_{ij} = \begin{cases} 
\frac{\pm i r}{2} & \text{if } j \text{ is odd} \\
\pm \frac{(2t-1) r}{2} & \text{if } j \text{ is even}
\end{cases} \quad \ldots \quad (\text{eq. 4.21}) \]

\[ y_{ij} = (j-1)\sqrt{15} \frac{r}{2} \quad \ldots \quad (\text{eq. 4.22}) \]

where \( i = 0, 1, 2, \ldots \) and \( j \) refers to the r-strip number = 1, 2, 3, 

Figure 4.14: Relocation of GCS as per Proposed Routing Algorithm

As stated, GCS will move parallel to r-strips and on way collect information from r-strip sensor. With the help of meta information available within GCS, shortest route will be decided. Here, in figure 4.14, data from node (4, 3) will be collected by GCS (3, 2) and passed on to node(3,2). Node (3, 2) will buffer the data till GCS (2, 1) arrives in its proximity, upon which the data is transferred to it. GCS (2, 1) will communicate the data to node (3, 1), and from here it travels through the r-strip and finally to base station. In generic, this data movement (→) may be represented as

\[ \text{node } (a, n) \rightarrow \text{GCS}(n, n-1) \rightarrow \text{node}(a-1, n-1) \quad \text{if } n \text{ is odd} \]

\[ \text{node } (a, n) \rightarrow \text{GCS } (n, n-1) \rightarrow \text{node } (a, n) \quad \text{if } n \text{ is even} \]
Since the data movement and distance traveled is maximum for nodes at lowest level i.e where n is of higher values, hence the movement of last GCS is initiated first.

(ii) Backward Data Collection

Once GCS travels to the r-strip node nearest to base station, reverse travel needs to be initiated. Here, the GCS instead of coming back in the same direction parallel to r-strip would exploit the mobility aspect of r-strip nodes. Figure 4.15 below represents a subsection of the complete deployed network, as depicted in Figure 4.14. Here GCS (3, 2), which had started its journey at time t=0, has reached near node (0, 3) at time t=k5.

Now at this point of time GCS instead of traveling back, is proposed to replace node(0,3), which replaces node(1,3), and so on, until node(5,3) occupies the position adorned by GCS(3,2) at time t=0. Hence, we then will have a new node acting as a GCS and the forward data collection phase executes recursively. The main significance of using this backward phase is that it reduces the activity performed by individual sensor acting as GCS, thereby increasing the lifetime of network. Moreover, it reduces the delay and forwarding cost.

Figure 4.15: Sub-section of deployed network to show reverse travel of GCS
Now, since all nodes are assumed to be equivalent, this goal-swapping makes no functional difference to the network. This resolution strategy is applied recursively, so that

\[ \text{node}(j-1, n) \sim \text{replaces} \sim \text{node}(j, n) \forall j \in 1, 2, 3 \ldots \ a \]

here, \( n \) is the \( r \)-strip number, while \( a \) is the offset of farthest node from base station in any given \( r \)-strip. The proposed mechanism executes in an iterative replacing successive nodes within given \( r \)-strip, and finally enabling the farthest node to occupy the initial position of GCS.

**B) Working Algorithm**

*Input required for the algorithm:*

- Value of \( n \), which is the number of \( r \)-strips deployed in subsurface
- Value of \( a \), which is offset of farthest node from base station in any given \( r \)-strip.
- Time \( k_i = d_i + t_i \) is the time for collector to travel from \( i \)th sensor to the \((i+1)\)th sensor (modulo \( n \) plus time required collect data from the \( i \)th head (\( ti > 0 \)).

1: INITIALISE
2: set \( q=a \) and \( w=n \)
3: set \( \text{time}=0 \)
4: Method: **transfer** \((q, w)\) \{start\}
5: GCS \((w, w-1)\) enquires data from node \((q, w)\)
6: \[ \text{if data \{node \((q, w)\)\} = true} \]
7: \[ \text{buffer data \{node \((q, w)\)\} to GCS \((w, w-1)\)} \]
8: \[ \text{if } [w \mod 2] = 1 \]
9: transfer data \{node (q, w)\} from GCS (w, w-1) to node (q-1, w-1)
10: \textbf{else}
11: transfer data \{node (q, w)\} from GCS (w, w-1) to node (q, w-1)
12: Method: \texttt{transfer (q, w)} \{end\}
13: find \( z = \text{time (modulo } ki) \)
14: if \( z \neq 0 \)
15: start a new thread \{ thread\_z \}
16: call \texttt{transfer (a, n-z)}
17: initialize movement of GCS (n, n-z)
18: GCS (n, n-1) arrives at node (a-1, n)
19: call \texttt{transfer (a-1, n)}
20: stop

\textbf{4.5 Conclusions}

The work presented in this chapter makes two unique contributions, initially towards the deployment of wireless sensors in subsurface and later proposes a query driven routing protocol as most of the eminent researchers have been silent towards the deployment of sensors and establishing communication among those. In addition, both the proposed strategies have proved to be efficient than their competent counterparts in other related fields. Further, the works exploit the mobility of sensors to great extent displacing normal sensors as well as GCS to formulate a new network layout, for data aggregation as well as dissemination. GCS’s which travel parallel to r-strips accumulate data on their way, and pass them over to next level r-strip, which definitely leads to a better route for the packets headed towards the base station.

Next chapter focuses on processing the information initially collected and later forwarded by GCS. The major aim would be to forward only relevant information
while filtering the fluctuations saving the energy and hence, improving the lifetime of network. Also, a robust and fault tolerant node replenishment mechanism has been proposed adding robustness to the proposed model.