HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Vishal Kumar Singh

# Node Selection for Cooperative Decoding in Wireless Mesh Networks

Masterarbeit im Fach Informatik

14. Januar 2019

Please cite as:
Vishal Kumar Singh, "Node Selection for Cooperative Decoding in Wireless Mesh Networks," Master's Thesis (Masterarbeit), Heinz Nixdorf Institute, Paderborn University, Germany, January 2019.

Distributed Embedded Systems (CCS Labs)
Heinz Nixdorf Institute, Paderborn University, Germany

Fürstenallee 11 · 33102 Paderborn · Germany

http://www.ccs-labs.org/

# Node Selection for Cooperative Decoding in Wireless Mesh Networks

Masterarbeit im Fach Informatik

vorgelegt von

**Vishal Kumar Singh**

geb. am 17. July 1992
in Kholara

angefertigt in der Fachgruppe

**Distributed Embedded Systems**
**(CCS Labs)**

**Heinz Nixdorf Institut**
**Universität Paderborn**

| | |
|---|---|
| Betreuer: | **Muhammad Nabeel** |
| Gutachter: | **Prof. Dr.-Ing. habil. Falko Dressler** |
| | **Prof. Dr. Christian Scheideler** |
| Abgabe der Arbeit: | **14. Januar 2019** |

# Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.
Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

# Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Vishal Kumar Singh)
Paderborn, 14. Januar 2019

# Abstract

We study the problem of gathering data and aggregation in decentralized, heterogeneous sensor networks for reliable communication. In many application scenarios, we have sensors with small energy budget and size in the network. Wildlife monitoring is one of many examples within the Internet of Things research community. To improve communication reliability and energy-efficiency of the network in such applications, macro-diversity has been employed on the data samples received by multiple sensor nodes in the network. Thus resulting in the reduction of transmission failures and further avoiding costly retransmissions. In recent times, macro-diversity techniques have been proposed which uses a distributed sensor network as an antenna array at the receiver end. These techniques primarily need the sensor nodes to forward the data samples to the sink node, to apply different diversity combining techniques. The process of forwarding the data samples from all the ground nodes at all times in the network incur a huge cost. We present two algorithms, a cluster and a tree-based one, that help to reduce the data transfers in the network by pushing the aggregation process near to the point of transmission within the network. Sensor nodes within the network act as an aggregator and apply diversity combining technique on the received samples from multiple receivers rather than at a centralized sink node. In an extensive set of simulations, we show that our algorithms substantially outperform naïve centralized solution and also depending upon the topology, cluster-based and tree-based algorithms outperform each other in terms of time delay and energy footprint.

# Contents

# Chapter 1

# Introduction

Wireless Sensor Networks (WSNs) have revolutionized habitat and environmental monitoring [1]–[3]. It allows us to study and monitor wildlife without any human intervention. WSNs usually have sensors with different capabilities. Tiny sensor node (referred to as a mobile node) is mounted on the targeted species[3]. These tiny sensor nodes continuously collect the data and transmit it to the static ground network on request. The ground network is made of many distributed sensor nodes (or ground nodes). These distributed sensor nodes have less energy restriction than the mobile sensor node and are connected to each other and a centralized node(referred to as sink node). Energy constraints of the mobile node posse challenge to the successful communication between the mobile node and other sensor nodes.

Sensor nodes use wireless communication to communicate with each other. Due to the low energy budget, the signal transmitted by the mobile node experience losses due to various channel effects like shadowing, multi-path fading and Free Space Path Loss (FSPL). Therefore, a sensor node may receive the signal from the mobile node but with high probability may not be able to decode it successfully, resulting in loss of data. To overcome such situation diversity combining technique has been proposed. Receive diversity helps us to exploit space diversity between sensor node, which treats these nodes as a distributed antenna array [4], [5].

To employ diversity combining, received samples needs to gather at a single node. The most naive approach can be to forward all the samples detected by the sensor nodes to the computationally superior sink node, followed by employing diversity combining on all the samples received at the sink node. Diversity combining can only help to successfully decode the signal if the channel losses experienced by different sensor nodes are uncorrelated to each other [6]. Employing diversity combining at the sink node is simple and effective; however, it has serious efficiency concerns. Forwarding of all the samples to the sink node from the sensor nodes surrounding the event will result in congestion, and increase in time to transfer the data to the sink node. The sensor nodes are acting as a relay will consume more energy to receive and transmit the samples to be forwarded. Consequently, resulting in a reduced lifetime of the WSN.

In the literature, some surveys exist [7]–[9], focusing on data aggregation techniques in the WSN. The literature survey focuses mainly on resource constraints like computational power, energy efficiency and security. Data aggregation aims at reducing the amount of data by performing suitable summarization. The aggregation techniques proposed in the literature have usually dealt with homogenous sensor networks, i.e., network with sensor nodes of similar capabilities.

In this work, we devise techniques to use channel resources and energy of the WSN efficiently, by finding an aggregator near the location of an event to perform data aggregation. Diversity combining is to be considered as an aggregation methodology, hence, producing a single stream of data to forward towards the sink node. Aggregation will result in less transmission towards the sink, therefore, resulting in efficient use of channel resources and energy of the sensor nodes in the network. We are considering a heterogeneous sensor network for our work, and it has three different sensor nodes, mobile node, ground node and a sink node. Mobile node has the highest energy restriction due to its small size, a ground node has comparatively lesser energy restrictions, and the sink node is computational superior and has the highest energy budget. Multiple ground nodes gather any data transmitted by the mobile node and forward it to the sink node.

To efficiently gather and forward aggregated data to the sink node we propose variants of cluster and tree approach. The proposed technique explores diversity combining as a way to reduce the forwarding of redundant data samples.

Our main contributions are be summarized as follows:

- We develop variants of cluster and tree algorithms for using diversity combining in a distributed fashion in the ground network.
- We use a wildlife monitoring scenario to investigate the performance of the algorithms proposed.
- We perform an extensive set of simulations and discuss the results.

We have submitted a conference paper "**Efficient Data Gathering for Decentralized Diversity Combining in Heterogeneous Sensor Networks**", for publication at IEEE WCNC 2019, based upon our work in the thesis. It is not yet published, as it is under peer review.

In the paper, I proposed and implemented the cluster-based algorithm, and I also implemented the centralized and the tree-based algorithm on the same model and application scenario to compare the three algorithms.

# Chapter 2

# Fundamentals & State-of-the-art

To understand the algorithms proposed and the results presented in the thesis, this chapter provides an overview of the background and state-of-the-art techniques. This chapter introduces WSN, effects observed in wireless channels and diversity combining techniques (Sections 2.1 to 2.3) We also present various efficient routing algorithms (Section 2.4). Additionally, we present an overview of BATS project and simulation tools for WSN (Sections 2.5 and 2.6).

## 2.1 Wireless Sensor Networks

WSN is a network made up of numbers of distributed sensor nodes. The sensor nodes are capable of detecting and gathering data. These sensor nodes are low power devices. They are fitted with *"limited memory, a power supply, a processor, a radio and an actuator"* [10]. With the help of onboard processors, sensors nodes can perform computation and forward partially processed data. These unique features make WSNs useful for a wide range of application areas. Akyildiz et al. [11] mention few application areas like *"health, military, weather monitoring, intrusion detection and monitoring in disaster areas"*.

The sensors nodes gather the data and often forward it to the sink node, i.e., more powerful node with more computation capability and act as a link to the internet. In WSN, nodes communicate with each other using a wireless link in an ad-hoc manner. This implies such networks do not need network manager. Nodes self-organize and manage the network. The sensor gathering data are also termed as the source node. Multiple sources nodes measurements are forwarded to the sink node, which is usually at a far distance. These source nodes forward the measured data by multihop architecture to the sink node.

Deployment environment of these sensor nodes makes its maintenance difficult and impractical. Sensor nodes are deployed to run on battery power, the use of battery power decides the lifetime of the node in the network. The energy consumption by a sensor node is attributed to three major activities carried out by such nodes i.e., monitoring or sensing, data processing and communication. In multihop architecture, sensor nodes also act as a relay for other source nodes further consuming energy in receiving and later transmitting the data towards the sink node. Thus reducing the number of such transmissions can help to extend the lifetime of the network.

## 2.2 Channel Losses

In WSN, sensor nodes use a wireless link to communicate to each other. Radio signals are propagating over the wireless link experience loss. These losses can be credited to effects like FSPL, shadowing and fading which radio signal experience, when in the air.

### 2.2.1 Free Space Path Loss

Consider a transmitter and receiver in a vacuum. Assuming the transmitter is an isotropic antenna, an isotropic antenna is defined as an antenna which radiates energy in all direction uniformly. The power of the signal transmitted by the transmitter antenna propagates equally in all direction.

The transmitter transmits the signal with a specific energy. The signal travels as a light wave in a spherical shape, with the transmitter at the centre. Sphere continuously grows moving away from the transmitter. The transmitted energy gets distributed equally over the sphere's surface.

The received power at the receiver depends on the size and orientation of the receiver to the transmitter. The receiver extracts power from a constant area of cross-section. As the distance from the transmitter increases the power per unit cross-section area decreases, resulting in less received power at the receiver.

This loss is expressed by Free-space path loss formula which is derived from Friis transmission formulae.

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{2.1}$$

The loss is directly proportional to the square of the distance $d$ from the transmitter and inversely proportional to the square of the wavelength $\lambda$ of the radio signal. But in real world vacuum does not exist.

The environment has trees, buildings, mountains etc. Signal experiences additional effects to the attenuation caused by the distance between transmitter and receiver. One of such effect is *blocking* or *shadowing* of radio signals due to large obstacles.

### 2.2.2 Shadowing

The radio signal propagates in free space in a straight line, like the light. Signals with higher frequency, behave more like the light. An even small obstacle like tree, building, or truck can block the signal completely. In extreme case, the radio signal is blocked entirely by obstacles.

Another effect is the reflection of radio signals. If the obstacle is substantial when compared to the wavelength of the signal, the radio signal is reflected. The radio signal attenuates on every reflection, as obstacle absorbs some of the signal's power. Reflection does attenuate the signal, but at times it helps the transmitting signals, in the case when the transmitter and receiver are not in straight line or no a line-of-sight (LOS) exists.

In case of the size of the obstacle being in the order of the wavelength or less, the radio signals get scattered. On scattering, the power of the transmitted signal gets scattered into many weak signals.

As we discussed radio signals propagate, like the light, it travels with the speed of light, but only in the vacuum. Its speed depends on the density of the medium. The radio signal that travels into the denser medium bents towards the medium, the effect is known as refraction. As the density of the atmosphere is higher closer to the ground, the LOS of the radio waves bents towards the earth.

### 2.2.3 Multipath Fading

Due to effects like scattering, reflection multiple copies of the signal reaches the receiver. Each copy travels a different path and experiences different attenuation, delay, or phase shift. The difference in phase and delay causes these copies of the signal to interfere either constructively or destructively at the receiver. Frequent destructive interference results in communication failure, this phenomenon is known as deep fading (i.e., Rayleigh fading).

## 2.3 Diversity Combining

Multiple copies of the radio signal available at the receiver experience more or less independent fading, if the radio channels are sufficiently separated in space, frequency, or time. The techniques to improve the effect of fading is known as diversity combining. These techniques use multiple copies of the signal to improve the Signal to Noise Ratio (SNR) of the received signal.

Transmit diversity and receive diversity are two popular schemes of diversity combining. Transmit diversity proposes to transmit the same symbol from multiple antennas and receive diversity employs multiple antennas at the receiver side to receive the symbol [12]. Employing combining to multiple copies of symbols helps to mitigate the fading effect in wireless communications without making any significant changes to the physical layer.

Scanning Diversity, Selection Diversity, Maximal-Ratio Combining (MRC), and Equal Gain Combining (EGC) are the common types of space diversity techniques that are used in practice [13]. Selection diversity proposes to pick the sample with the highest SNR of all the samples and use that alone to decode the signal. EGC is the simplest of all the techniques as it assigns equal gain to all the diversity branch and adds merely all the samples from different branches. MRC is a more sophisticated technique as it calculates the gain of every diversity branch, which is proportional to the rms signal level and inversely proportional to the mean square noise level and than adds constructively all the copies of the signal. A detailed comparative study of all these combining techniques presents MRC to be the best-combining technique and EGC provides the best performance only for low bit-error rate (BER) values [14].

In Space diversity, if antennas are present too near to each other and there is no sufficient spacing between them then; as a result, all antennas experience correlated fading affecting the overall diversity gain. According to [12], to experience uncorrelated channels in a two-branch receiver, the antennas must be separated in space in the order of ten wavelengths. Effect of correlated Nakagami fading on MRC is studied in [15]. It turns out that the system performance deteriorates as the correlation increases between the branches. Applying diversity combining on distributed systems provide higher robustness against fading and shadowing effects [16], thus, resulting in higher diversity gain. But performing diversity in a distributed network requires complex receivers.

Some wireless networks usually comprise of low power receivers or nodes and every receiver does not have a capability to perform complex operations to accomplish diversity combining. Such receivers can act as a relay to some other stronger receiver using schemes such as Amplify-and-Forward (AF) and Decode-and-Forward (DF) [17], and thus enable cooperative diversity [18].

WSNs have seen a rise in their use to monitor animals in the wild. BATS project [19] is one such attempt, which uses diversity combining and looks into the practical aspects of diversity combining [5] in ultra-low power WSNs.

Performing diversity combining helps us to mitigate fading losses. Employing diversity combining needs samples to be gathered at a single point.

## 2.4 Data Gathering Algorithms

In a WSNs, sink node is critical to the operation of the network. As sink node act as a gateway to the outside network for other sensors in the network. Death of the sink node will result in the collapse of the functionality of the whole network. According to Anastasi et al. [20], communication is a significant factor of energy consumption in WSN. To increase the lifetime of the sink node and improve the energy efficiency of the network, efficient data aggregation algorithms are employed within the sensor network, which in turn reduces the number of transmissions made within the network. For data aggregation algorithms to work, data needs to be efficiently collected, and these gathering algorithms can be categorized as the cluster and tree-based algorithms.

### 2.4.1 Cluster based algorithm

In a WSNs, neighbouring sensors have a similar perception of an event. Thus clustering sensor nodes help in reducing the transmission of redundant data, and therefore increasing the lifetime of the network. In the past [11], algorithms have used random selection to elect Cluster Head (CH), some have taken a more informed approach where the residual energy of the sensor nodes or the distance of nodes from the sink nodes has been used to elect CH.

In past, few popular clustering algorithms have been proposed for WSN [21]–[24]. These algorithms were simple but did not provide the most efficient solution. The major drawbacks were a non-uniform distribution of cluster heads, resulting in non-uniform energy consumption in the network.

Random competition based clustering (RCC) [25] is proposed for mobile ad hoc networks competition based clustering technique, which could also be used for WSN. It is a distributed clustering algorithm, which is simple and focuses on the stability of clusters. In the network, the node declaring first wins and becomes a CH and govern the rest of the nodes in its radio range. Nodes in the radio range on listening to the broadcast give up their opportunity to be a CH and accept the first node as the CH. In such distributed setup there is the probability of nodes announcing their candidature as CH to be concurrent. Nodes use a random timer to resolve any such conflict. Every node before transmitting its CH claim reset its random timer, and if it receives another CH claim in this time, it ceases to transmit its claim to be CH. On selection of CH, it can act as an aggregator of the data gathered by the Cluster Member (CM).

Ding, Holliday, and Celik [26] proposed a Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC), to ensure uniform distribution of CH in the network and achieve less energy consumption. It works in a distributed manner wherein every node calculates its weight once it locates all the neighbours in the sensor network. The weight is the function of the energy and the neighbours. The node with the highest weight become the CH, and the other nodes in the neighbourhood become the CM. The nodes in the cluster are called first level members, and they keep adjusting its path to reach the CH using the least energy path via other nodes. The new path discovered may be multi-hop, and nodes can be at the second level, to limit the number of levels range is specified.

Usually, the clustering algorithm focus on the generation of a minimum number of independent clusters. Youssef et al. [27] proposed a technique Multi-hop Overlapping Clustering Algorithm (MOCA) in which they argued that having some level of overlapping can help in routing localization and recovery in case of CH failure. Each node becomes CH which some probability p, after that each node advertise itself to all the nodes in the network in its radio range. The nodes forward the CH advertisement to a predefined number of hops. Nodes in the sensor network send a request to all the CH it has heard from to join the cluster. Node ID of all the CH is sent with the request, which in turn helps to identify the number of the cluster the node is part off. The probability p helps to control the count of clusters in the network.

Another energy-aware protocol for nonuniformly distributed and heterogeneous nodes was proposed by Yu et al. [28]. Energy consumption of the network is uniformly distributed in the network by increasing the work to the CH in the sparse areas. Avoid Near Cluster Heads (ANCH) [29], is a comparatively new algorithm. It also focuses on the uniform distribution of CHs in the network. First, all the potential CHs are identified. Then elimination of CHs which are close to each other starts to reach an optimum number of CH.

### 2.4.2 Tree based algorithm

Tree-based algorithms [30], [31] usually rely on the hierarchical structure of the nodes in the network, rooted at the sink node in the network. Tree algorithm relies on the construction of a spanning tree rooted at the sink node to perform data gathering and aggregation. The simple and efficient way of data aggregation is to define the direction of the flow of the data and mark intermediate nodes as an aggregator within the network.

Ding, Cheng, and Xue [32] proposed a heuristic to construct and maintain an aggregation tree. The non-leaf nodes are responsible for data aggregation and data forwarding. They considered residual energy as important heuristics. Higher residual energy of a sensor node gives it a higher chance to be a non-leaf node, broadcast first and also be a parent to a node in case child node has two parents. During aggregation tree construction every node transmits only once. A tree is constructed periodically if residual power is below some threshold, its children node change to new parents.

Kuo and Tsai [33] studied the problem of constructing energy efficient data aggregation trees. They showed that this problem is NP-complete. Shortest path tree algorithm and Steiner tree algorithm was probed by them and they found that both have bad approximation ratio, so they proposed a new approximation algorithm in which they considered the wireless links with low quality and showed proposed algorithm perform well in terms of energy cost.

Recently Lin and Chen [34] presented a study focusing on the construction of maximum-lifetime data aggregation tree in WSN. They proposed an approximation algorithm to construct a data aggregation tree, whose inverse lifetime is guaranteed to be within a bound from the optimal aggregation tree. They have shown that the adjustable transmission power of the sensor nodes has advantages over fixed transmission power of the sensor nodes to maximum levels in the network. Thus they were able to increase the lifetime of the network.

## 2.5 BATS Project

BATS project relies on ultra-low power sensor nodes which are carried by the bats to monitor the movement of the bats in the wild [19]. These small and light weight sensor nodes weigh only 2 g so that they are easily mountable on the bats with an average weight of 20 g. When these bat nodes come in contact with the other bat nodes, they store the contact information. Later, when they visit the hunting ground, which happens on an irregular basis, the wake-up receiver of the bat node is triggered by the ground nodes whenever in range. Because of the foliage environment, the signal received at the ground nodes is affected by multi-path fading and shadowing. The highly mobile nature of the bat nodes also make the communication difficult. Energy constraint of the bat node prohibits the nodes from using standard approaches like the repeated transmissions. Other approaches like Forward Error Correction (FEC) using fountain codes was also investigated in [35] to improve the reliability which provided encouraging results but not sufficient enough.

A novel technique of soft-bit diversity combining was developed and investigated in [36]. It used distributed nodes as collaborators to perform diversity combining and improve the Packet Delivery Rate (PDR) at the ground nodes. As this approach converted the received signal into soft-bit values before diversity combining, information was lost and affected the overall diversity gain. A new technique was proposed which exploited the signal-level diversity in the same distributed network [5]. It forwarded the selected signal samples to the central node, which performed synchronization and phase correction before applying diversity combining techniques on the signal samples resulting in maximum diversity gain.

As stated before, if diversity combining is performed at a central node, every node acting as a branch needs to forward the received data to the central node, however, in the case of distributed diversity combining, it needs to be performed in one of the involved nodes. To enable distributed diversity combining, nodes need to cooperate with each other.

## 2.6 Simulation of WSN

With the increase in the applications of WSNs, the complexity and the scale of such networks are increasing. Thus, studying the behaviour of an individual sensor node and WSN becoming complex. Therefore, to conduct a study of such large and complex application, simulation is used. Researchers rely heavily on simulation to verify their ideas. Simulation enables rapid prototyping of sensor nodes of WSN and protocols using a high-level programming language like C++. It also makes it easy to debug and visualize the protocols behaviour employed in the network.

There are few well-known simulators used to simulate WSN, such as OMNeT++[1] and ns-3[2]. Both the simulators are a discrete-event simulator.

OMNeT++ is modular, component-based network simulation library written in C++ [37]. It is an open source library and free to use for non-commercial purpose. OMNeT++ is a simulation framework not a simulator on its own. It does not contain models for network protocol like IP. The external framework developed for OMNeT++ makes it an attractive option for network simulation. It is considered to be better than other network simulators [38]. The frameworks developed for OMNeT++ makes it a desirable option to use for simulation.

INET *framework*[3] is one such framework developed for OMNeT++. It is also an open-source network simulation package. INET contains models of the internal stack (TCP, UDP, IPv4, etc.), it also includes the model for wired and wireless protocols (Ethernet, IEEE 802.11, etc.), supports mobility models, physical layer and many other protocols. INET built around the concept of modules like OMNeT++, and message passing is used to communicate between different modules. The small components and network protocols can be used to form custom hosts, routers, switches or wireless nodes. The modular structure of INET makes it easy to understand the existing components and protocols as well as provide an opportunity for users to define and implement their components and protocols to validate new ones. INET uses the infrastructure provided by the OMNeT++ to execute the simulations from IDE or command line, parametrize the simulations or record data generated to study or verify the behaviour of protocols.

[1] `https://www.omnetpp.org/`
[2] `https://www.nsnam.org/`
[3] `https://inet.omnetpp.org/`

# Chapter 3

# Data Gathering Techniques

There have been techniques proposed in the past to gather and aggregate data in an efficient way and send it to the sink node, such as cluster approach and tree approach.

Clustering algorithms and tree algorithms have been summarized by [7], [39]. Clustering algorithms can be categorized into two categories of centralized and distributed algorithms [40]. In distributed clustering algorithms, all the decision related to cluster formation is taken by sensor nodes based on its internal information. In contrast, centralized clustering algorithms rely on a central node or sink node for the decision making. The sink node has a complete view of the whole network.

In BATS scenario, we have a heterogeneous WSN. Bat with a sensor act as a mobile node. Sensor nodes or ground node which are static, communicate with each other via wireless communication. Sink node acts as a gateway for the WSN, and it doesn't have the view of the whole network from the outset.

## 3.1 Problem Statement

Data samples received by ground nodes in the network has to be gathered at one point for employing diversity combining, which allows us to decode signals at the receiver. Before gathering data, we need a node which can act as an aggregator and then the diversity combining technique can be employed.

If the point of aggregation is far from the Point-of-Origin (POO), it is the position at which the mobile node makes the transmission; all the copies will have to be forwarded to the node chosen to be an aggregator. Each transmission and reception costs energy to the ground nodes and utilizes the wireless link. Thus the lifetime of the sensor nodes in the network depends on the point at which aggregation is performed. If the aggregator is far from the POO, we will need more transmission to gather the data at the aggregator, thus consuming more energy of the sensor nodes, which are energy constraint.

This problem can be seen as an exercise to select an aggregator near to POO and thus pushing the point of aggregation within the network. We cannot choose a static aggregator as our source is mobile, resulting in different POO after every transmission. The mobile nature of the source compels us to select a new aggregator after every transmission.

Choosing an aggregator near the POO will result in less transmission to gather the data and assure efficient usage of the limited energy of the sensor networks and increasing the lifetime of the sensor nodes in the network.

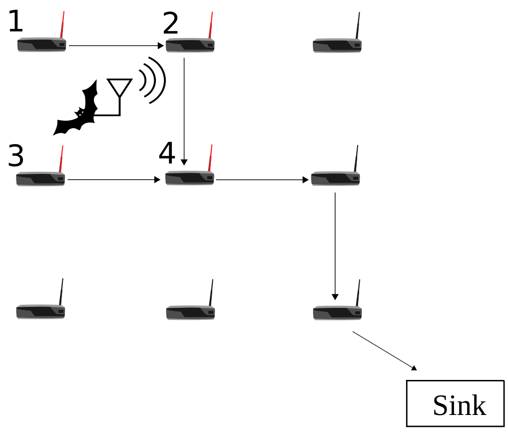



**Figure 3.1** – Mobile bat transmitter, Sink node and distributed ground nodes that detect the transmitted signal.

In order to understand it better consider Figure 3.1. Nodes marked as 1, 2, 3, 4, are considered to have the sample from the mobile node. Each node is at non-zero hop distance from the sink node.

$$\text{Node } 1 \rightarrow 5$$
$$\text{Node } 2 \rightarrow 4$$
$$\text{Node } 3 \rightarrow 4$$
$$\text{Node } 4 \rightarrow 3$$

If we perform diversity combining at node 4, we will reduce the number of transmissions as node 1, 2 and 3 hop count distance to node 4 is less. Hence only node 4 will have to transmit the combined data stream to the sink node.

$$\text{Node } 1 \rightarrow 2$$

$$\text{Node } 2 \rightarrow 1$$
$$\text{Node } 3 \rightarrow 1$$
$$\text{Node } 4 \rightarrow 3$$

In our scenario, ground nodes are unaware about the other nodes which have the copies of the sample. We need to define techniques which allows them to cooperate with each other using minimum overhead.

## 3.2 Centralized Approach

Our first, naive approach is a centralized approach. In this approach (cf. Algorithm 3.1), sensor node on detecting samples transmitted by the mobile node forwards it towards the sink node. Every sensor node cannot communicate with the sink node directly. Thus other sensor nodes are used to communicate to the sink node. In this multihop setup, other sensor nodes act as a relay and forward the data sample to the sink node. The sink node on receiving samples from various sensor nodes act as an aggregator and employ diversity combining. Once combining is performed, the sink node tries to decode the combined signal to recover the data transmitted by the mobile node.

**Require:** event ∈ {signal from mobile node, signal from ground node}
**Ensure:** received signal is forwarded to sink

**switch** (event)
**case** signal from mobile node**:**
**case** signal from ground node**:**
  **if** `currentNode` is `sink` **then**
    employ diversity combining with the signal copies received and, afterwards, decode
  **else**
    forward the received signal to sink
  **end if**
**end switch**

**Algorithm 3.1** – `Centralized`

This approach act as a baseline for comparison and measuring the performance of other approaches. As sink node act as an aggregator, gathering data from all the sensor nodes receiving the data at the sink node becomes vital to decode the signal with high probability. The number of transmissions done is equal to the hop distance of the sensor node from the sink node.

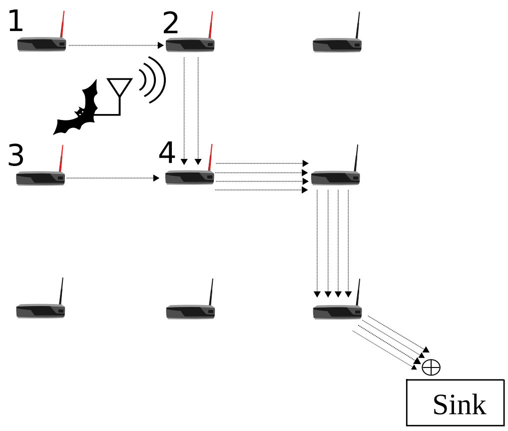



**Figure 3.2** – Mobile bat transmitter, Sink node and distributed ground nodes that detect the transmitted signal. The addition function shows where diversity combining is employed. Arrows indicate necessary transmissions for centralized approach.

Consider an example in Figure 3.2. The nodes marked as 1, 2, 3 and 4 receive sample from mobile bat transmitter. For samples to reach sink node from respective sensor nodes take the following number of transmissions:

$$\text{Node } 1 \rightarrow 5$$
$$\text{Node } 2 \rightarrow 4$$
$$\text{Node } 3 \rightarrow 4$$
$$\text{Node } 4 \rightarrow 3$$

After the number of transmissions specified above is made, all data sample will reach the sink. Subsequently, sink node can perform diversity combining on the received samples.

## 3.3 Cluster Approach

In the cluster approach, we want to push the process of aggregation within the network by selecting an aggregator near POO. Thus to answer the question of designating an aggregator and gathering data at the aggregator in a distributed network, we take inspiration from the standard method of clustering.

In a cluster, we have a CH and CM, wherein CM sends data to CH. Thus solving our problem of gathering the data and designating an aggregator, asCH can act as an aggregator by employing diversity combining technique on the data samples sent by all the CM.

**Require:** event ∈ {signal from mobile node, signal from ground node, *cBackoffTime* expired, *slaveBackoff* expired}
**Ensure:** received signal is forwarded to sink

**switch** (event)
**case** signal from mobile node**:**
  **if** received SNR > $SNR_{max}$ **then**
    *cBackoffTime* ← 0.0
  **else**
    map received SNR on the scale of $SNR_{diff}$ (calculated from $SNR_{max}$ - $SNR_{min}$)
    *cBackoffTime* ← wait time from the scale
  **end if**
  start *cBackoffTime*
**case** *cBackoffTime* expired**:**
  broadcast CH selection and start *slaveBackoff*
**case** signal from ground node**:**
  **if** `currentNode` is `sink` **then**
    employ diversity combining with the signal copies received and, afterwards, decode
  **else if** CH selection broadcasted **then**
    forward received signal copy to CH
    cancel *cBackoffTime*
  **else**
    forward the signal to sink
  **end if**
**case** *slaveBackoff* expired**:**
  employ diversity combining with the signal copies received, decode, and forward the result to sink
**end switch**

**Algorithm 3.2** – `Cluster`

We propose Cluster Algorithm 3.2, a variant of standard clustering technique. For the cluster to form the sensor nodes receiving samples from the mobile node needs to cooperate. On receiving the data from the mobile node, sensor nodes use SNR of the received signal in the arbitration to become CH. All the sensor nodes with the sample, start a *cBackoffTime* based on the received SNR. The sensor node with highest SNR gets the role the CH and also the aggregator. The CH broadcast its selection on winning the arbitration and wait for a *slaveBackoff* for other sensor nodes with a copy of the signal to forward the copies. Other sensor nodes with the copy of the signal, act as a CM and forward the copy of the signal from the mobile node to the selected CH. On expiry of *slaveBackoff* timer, CH employs diversity combining on all the samples gathered from the CM, resulting in a single stream of decoded data from all the gathered samples. Finally, combined data is forwarded to the sink node. The CH may be more than a hop distance away from sink node, hence other nodes within the network act as a relay for the CH.

In the algorithm, $SNR_{max}$ and $SNR_{min}$ represent configurable SNR thresholds, which define the scale on which receives SNR can be mapped. Thus helping on setting *cBackoffTime*, which defines the time sensor node to wait before sending the CH selection message. CH also employs *slaveBackoff* time, which is the waiting time at the CH for receiving samples before attempting diversity combining on the received samples.

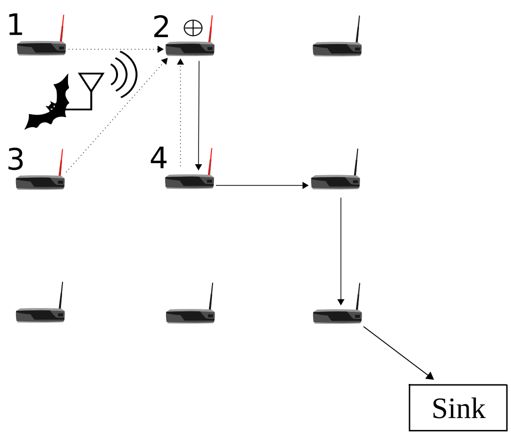



**Figure 3.3** – Mobile bat transmitter, Sink node and distributed ground nodes that detect the transmitted signal. The addition function shows where diversity combining is employed. Arrows indicate necessary transmissions for cluster approach.

Consider example in Figure 3.3. The nodes marked as 1, 2, 3 and 4 receive a sample from mobile bat transmitter. In this case, all nodes do not forward the samples directly to the sink node. Sensor node 2, becomes CH and other nodes forward the samples with them to sensor node 2 indicated by dotted arrows in the figure. Then diversity combining is applied at node 2, indicated by an addition function symbol. After decoding the data, it is forwarded to the sink node which is indicated by solid arrow lines in the figure.

This approach has a small overhead of forming the cluster, but it has a clear advantage as it takes less transmission for decoded data to reach the sink node.

## 3.4 Tree Approach

Tree approach like cluster approach tries to push aggregation in the network instead of performing it at the sink node. All the nodes in the ground network form a tree, rooted at sink node. The nodes in the network are assigned a level, and multiple nodes can be at a level. In this approach, nodes are not marked as an aggregator explicitly unlike that of the cluster approach. The nodes on receiving data from the mobile node wait before forwarding the data towards the sink node. Thus allowing the data from a node at a higher level to trickle down and enable the node to act as an aggregator by employing diversity combining on the samples received and forward the samples as a single stream of data towards the sink node.

**Require:** event ∈ {signal from mobile node, signal from ground node, *tBackoffTime* expired}
**Ensure:** received signal is forwarded to sink

```
switch (event)
case signal from mobile node:
  tBackoffTime ← level ∗ baseBackoffTime
  start tBackoffTime
case signal from ground node:
  if currentNode is sink then
    employ diversity combining with the signal copies received, if still not decoded
  else if signal already decoded then
    tBackoffTime ← 0.0
    start tBackoffTime
  else
    tBackoffTime ← baseBackoffTime
    start tBackoffTime
  end if
case tBackoffTime expired:
  if received more than one copy of the same signal then
    employ diversity combining
  end if
  forward the signal to sink
end switch
```

**Algorithm 3.3** – `Tree`

We propose Tree Algorithm 3.3, in which sensor nodes on receiving the data from the mobile node, compute *tBackoffTime* timer. *tBackoffTime* is computed as product of its *level* and *baseBackoffTime*, where *level* represents the sensor node position relative to the sink node. If the sensor node receives multiple copies before *tBackoffTime* expires, nodes employ diversity combining before forwarding the data towards the sink node. The sensor node which does not receive the sample from the mobile node, but receives a signal for another sensor node, waits for one *baseBackoffTime* before forwarding the received signals. If during the wait multiple signals are received, it also acts as an aggregator and employs diversity combining on the samples.

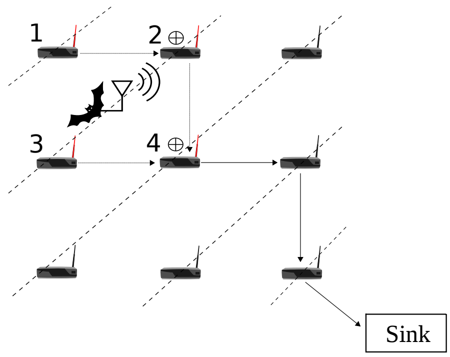



**Figure 3.4** – Mobile bat transmitter, Sink node and distributed ground nodes that detect the transmitted signal. The addition function shows where diversity combining is employed. Arrows indicate necessary transmissions for tree approach.

Consider example in Figure 3.4,the nodes receiving the sample from the mobile node are marked with 1, 2, 3, 4. The node 1 is farthest from the sink, and it is on *level* ← 0, node 2 and 3 are at same *level* ← 1 and node 4 is at *level* ← 2. Nodes 1,2,3 and 4 on receiving the sample from the mobile node calculate the *tBackoffTime*. Node 2 will wait for one *baseBackoffTime* more than that of node 1, thus node 2 will get the message forwarded by node 1 and will be able to employ diversity combining on the samples received and forward only one single stream forward. This wait mechanism allows aggregation to happen within the network and doesn't call for an explicit marking of nodes as an aggregator.

## 3.5 Model Implementation

We implemented the algorithms discussed in the previous sections in the INET *framework*[4]. INET provides an implementation of realistic wireless channel models, functionality for various ISO layers and mobility models.

[4] `https://github.com/inet-framework/inet`

### 3.5.1 Application Scenario

To evaluate our data gathering approaches, we have designed a scenario based on BATS project. The network is heterogeneous in nature. It consists of a mobile sensor node mounted on the bats (mobile node), static sensor nodes (ground nodes) and a sink node.

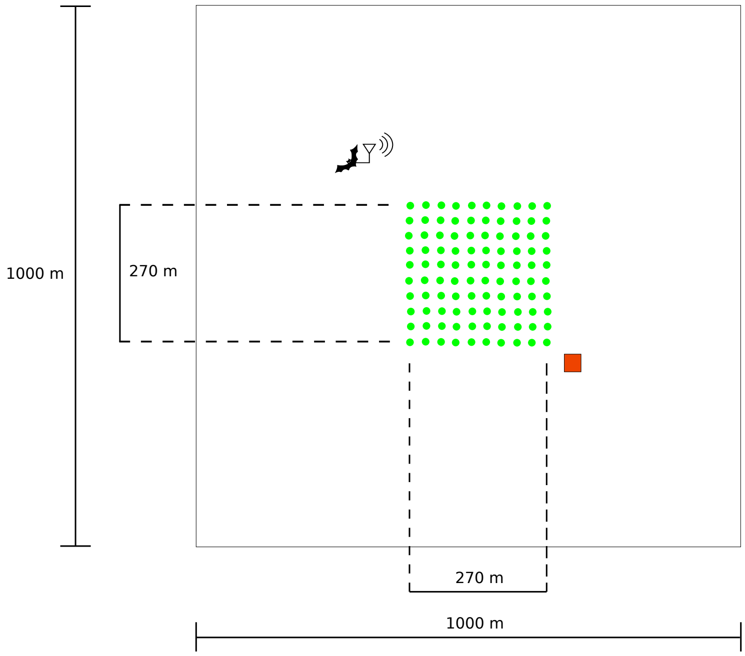



**Figure 3.5** – BATS simulation network.

We have designed a two-dimensional model as discussed in [35]. It has a system area of 1000 m × 1000 m, which has a hunting area consisting of 100 nodes in the centre as shown in Figure 3.5. The ground nodes are marked in green color and the sink node has been marked in red color. The nodes are placed in a grid, with inter-distance of 30 m. Further, we have used two different positions for the sink node i.e., at the centre and at the side of the network. To model the mobility of bat we have used Random Waypoint model [41]. In literature [42], the harmful effects of random waypoint mobility have been discussed. Hence to have a uniform distribution of transmission from the mobile node over the hunting area, we place the ground nodes at the centre of the network. We also assume that nodes in the network are synchronized up to a level of ms using Network Time Protocol (NTP) [43]. Nodes communicate with each other using standard WiFi protocol. To simulate realistic wireless channel, we have included Additive White Gaussian Noise (AWGN), FSPL, and fading loss of 0.25 dB/m [3].

In our model, for communication between ground nodes in the ground network, we have defined static routing. Static routing is shown in Figures 3.6 and 3.7. Circles represent ground nodes and rectangle represent the sink node in the network. The arrow indicates the next hop from the current node.

### 3.5.2 Cluster Formation

In the centralized algorithm, the ground nodes receiving a data sample from the mobile node forward it towards the sink node. Centralized algorithm considers the sink node to be the centralized aggregator. The diversity combining is performed, once all the data samples are received at the sink node.

Cluster algorithm takes a very different approach when it comes to the position of performing aggregation. The most significant shift it does is moving the process of aggregation within the network near to the POO. Cluster algorithm does not

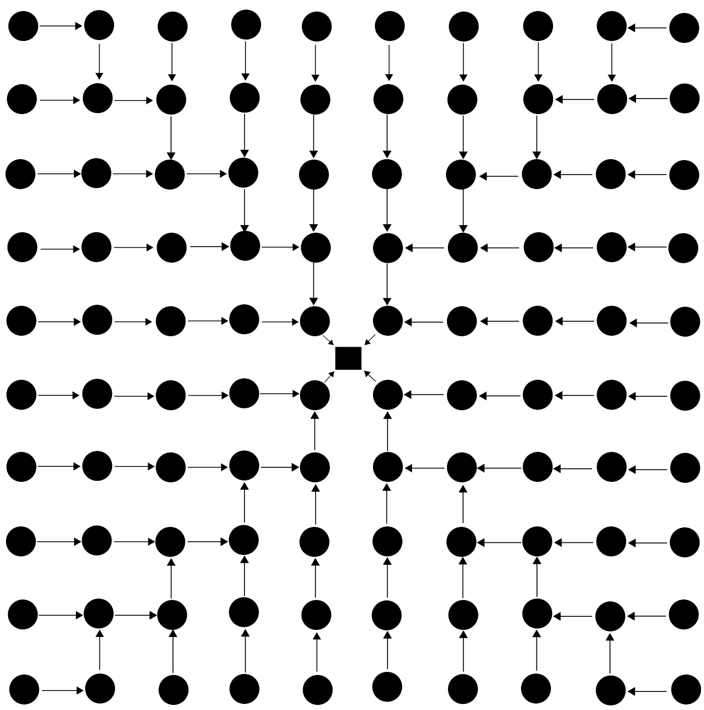


**Figure 3.6** – Static routing with the sink node at centre of the hunting area.

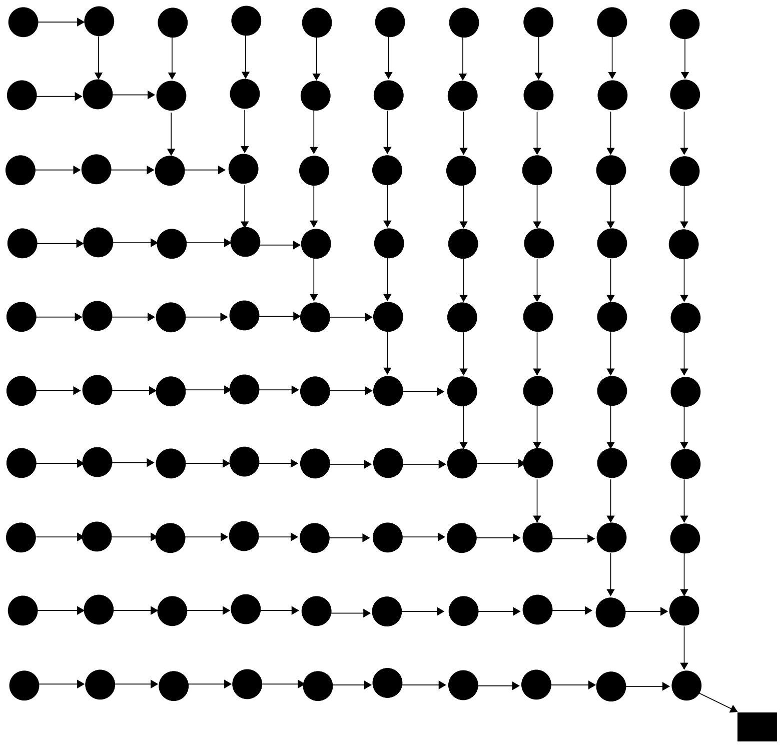


**Figure 3.7** – Static routing with the sink node outside the hunting area.

consider the sink node to be the only aggregator in the network rather CH of every cluster also act as an aggregator. We can also see the process of cluster formation as the process of data gathering.

As ground nodes in the ground network do not know which all nodes receive the signal from the mobile node, therefore, a distributed technique is used to form the cluster. The ground nodes receiving the data sample from the mobile node use the SNR of the collected data sample, to compete with each other to become the CH of the cluster. Ground nodes independently decide the wait time before announcing their selection as a CH. Intuitively, wait time of any ground node is inversely proportional to the received SNR. The node with the highest SNR will announce its selection as CH before than any other ground node. It can also be said as the node nearest to the mobile node at the time of transmission has the highest probability to become a CH as that node with high probability will see lesser channel losses.

Received SNR provides us with a great way to resolve contention, but to effectively use SNR to define wait time we need to establish a scale to set bounds on the wait time. Establishing bounds on the wait time will allow us to finish the process of cluster formation under stipulated time. In our case answer comes from the topology of the ground network. In our model, FSPL is the major contributor to the channel losses as seen in the outdoor measurements [44]. Hence, it enables us to estimate the losses experienced by the samples before reaching a ground nodes at a certain distance. Geometrically looking at the ground network, set of four nodes form a square of side 30 m length. We define two arcs of radius max-radius and min-radius respectively, around every ground node shown in Figure 3.8. While transmitting the samples, the mobile node can be in the area under the curve formed by max-radius of only one ground node at any instance, hence the ground node having received SNR greater than the $SNR_{max}$becomes CH, resulting in avoidance of any kind of contention with other ground nodes for CH selection. $SNR_{max}$ (in dB) is calculated as

$$SNR_{max} = P_{\text{tx}} - L_{\text{noise}} - L_{\text{FadingMax}} - L_{\text{FreeSpaceMax}}, \tag{3.1}$$

where $P_{\text{tx}}$ is the transmit power and $L$ represents the different loss terms. $SNR_{max}$, $L_{\text{FadingMax}}$, $L_{\text{FreeSpaceMax}}$ are values calculated at the point max-radius which is 14 m in our scenario. The ground nodes having received SNR less than $SNR_{min}$ can never

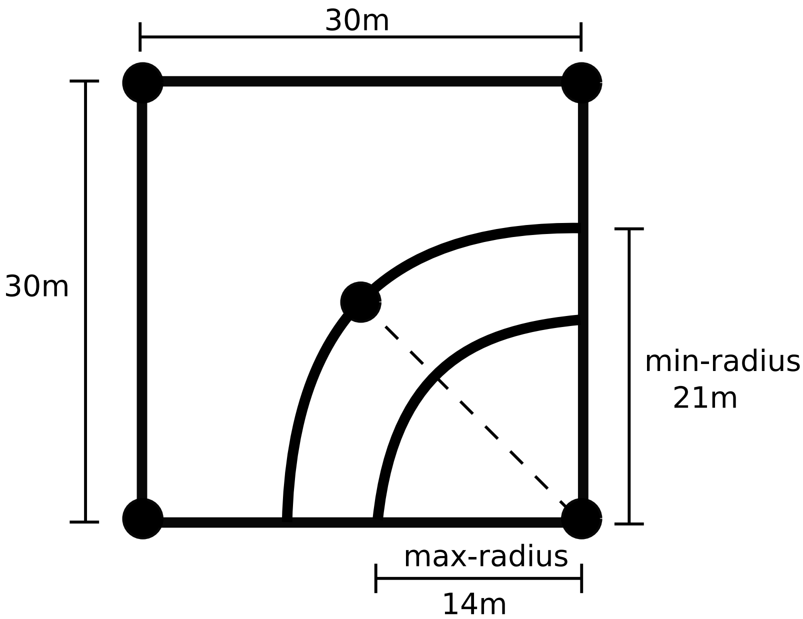



**Figure 3.8** – A simplified sub-grid with inter-distance of 30 m between nodes showing arcs formed by max-radius and min-radius.

be selected as CH, where $SNR_{min}$ (in dB) is calculated as

$$SNR_{min} = P_{\text{tx}} - L_{\text{noise}} - L_{\text{FadingMin}} - L_{\text{freeSpaceMin}}, \tag{3.2}$$

$SNR_{min}$, $L_{\text{FadingMin}}$, $L_{\text{FreeSpaceMin}}$ are values calculated at the point min-radius which is 21 m. Similarly, if the SNR is less than $SNR_{max}$ but greater than $SNR_{min}$, the ground nodes will take part in the contention to be CH. We define a scale of length $SNR_{diff} = SNR_{max} - SNR_{min}$, the scale is divided in discrete parts. We assign back-off time to each part. Every contender node finds *cBackoffTime* (in ms) by mapping its received SNR on the scale of $SNR_{diff}$ as shown in Figure 3.9.

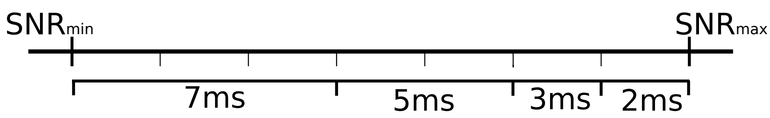



**Figure 3.9** – Scale mapping received SNR to find *cBackoffTime*.

As the difference between two points max and min points is 7 m. We divide the $SNR_{diff}$ into seven equally sized slots. The scale initially is finely divided, and the first two slots get wait time as 2 ms and 3 ms respectively, whereas rest of the slots are coarsely divided, 3rd and 4th slots are assigned wait time of 5 ms and rest of the slots are assigned wait time of 7 ms. The idea behind doing so is to identify the node nearest to the mobile node as quickly as possible.

The ground node with least *cBackoffTime* expires first and announces to be CH. On listening to the broadcast, the ground nodes join the cluster as CM and forward the sample from the mobile node to the CH. The CH waits for *slaveBackoff* time after the broadcast, to receive samples from the CM. The expiry of *slaveBackoff* at CH, is followed by employing diversity combining on all the samples received. We have set *slaveBackoff* time to 10 ms in our implementation as the ground network can receive at most one transmission per time interval according to the BATS protocol [3].

### 3.5.3 Tree Formation

In our tree approach, we define the *level* to which each ground node belongs, levels are represented by dotted lines in Figure 3.4. The node present on the one-dotted line belongs to the same level. The level value increases as we move towards the sink node.

Every ground node on receiving signal sample transmitted by the mobile node initializes a *tBackoffTime* timer. The value of the *tBackoffTime* timer directly depends on the value of *level* and *baseBackoffTime*. On expiry of *tBackoffTime*, the respective ground node checks if it has received any copies from ground nodes at a lower level, further, it performs diversity combining on the received samples. Finally, combined data is forwarded towards the sink node. The value of *level* (starting at integer value zero) increases as we move towards the sink node. Every node in the network is a potential aggregator, and sensor nodes near to that of the sink node have more probability of becoming an aggregator. As multiple sensor nodes get data from the mobile node and forward their data samples independently unlike the cluster approach, we will have multiple aggregators in the network on the way to the sink node. Other ground nodes present in the way to the sink node wait for *baseBackoffTime* before forwarding the data towards the sink. Hence value of *baseBackoffTime* is a critical value.

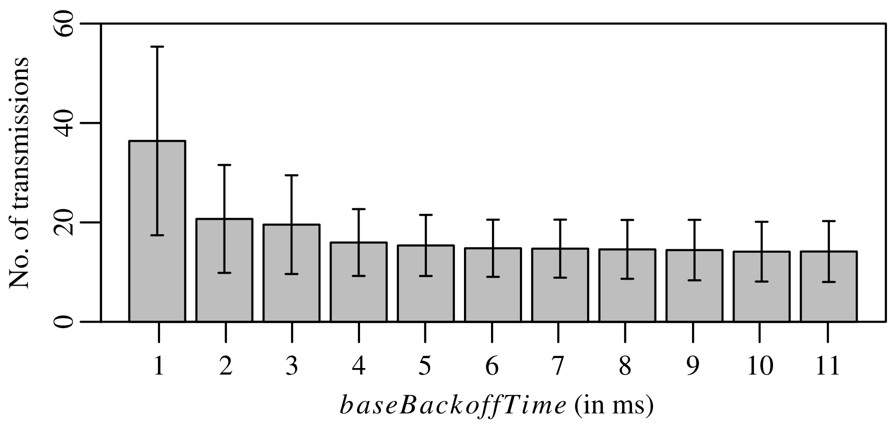



**Figure 3.10** – Parameter study to identify optimal *baseBackoffTime* in Tree approach.

If we choose *baseBackoffTime* value too small, the nodes will forward the signal towards the sink node before the sample from the ground nodes at lower level arrive; if it is too high end-to-end time delay will increase. Thus *baseBackoffTime* value becomes a configurable parameter in this approach, which needs to be tailored as per the scenario.

To identify the optimal *baseBackoffTime*, we conducted a parameter study with various *baseBackoffTime*. In Figure 3.10, we recorded the number of transmissions required to make data available at the sink node for *baseBackoffTime* from 1 ms to 11 ms. The error bars indicate the standard deviation in the plot. We can see in that after 6 ms reduction in the number of transmission stops. Hence we choose 6 ms as the *baseBackoffTime* value for our simulation.

# Chapter 4

# Evaluation

We evaluated our both cluster and tree approach with the help of extensive simulations done using INET *framework*. We have simulated both the approaches for sink position at centre and outside the network. As a baseline approach we have also evaluated a centralized approach. We have used multiple parameters to study the effectiveness and difference between our approaches.

## 4.1 Simulation Setup

We used BATS application for our simulation scenario. Applications scenario has been explained in chapter 3. A mobile node (i.e., Bat) starts from the left - top corner in the setup. In our setup bat uses random waypoint mobility model. Mobile node transmits sample every 100 ms. The size of the packet transmitted by a mobile node is 12 B [5].

We have considered only one mobile node in our simulation. We have conducted our simulation for two different positions of the sink node in the network. Performance of approaches has been studied with the sink node at the centre and outside

| Parameter | Value |
|---|---|
| Protocol | IEEE 802.11 |
| Simulation Time | 115s |
| Repetitions | 20 |
| Ground Node Count | 100 |
| Mobile Node Count | 1 |
| Sink Node Count | 1 |
| Energy | 50nJ/bit |

**Table 4.1** – Simulation parameter study values

| Component | Parameter | Value |
|---|---|---|
| Mobile Node | Transmission Power | -5dBm |
| | Packet length | 12B |
| Ground Node | Transmitter Power | 5dBm |
| and | Sensitivity | -85dBm |
| Sink Node | Bitrate | 24Mbps |
| | Packet Length | 3840B |

**Table 4.2** – Simulation parameter for mobile node, ground node and sink node

at the network. Sink node outside the network helps us evaluate the worst case end-to-end time delay of our approaches.

We have 100 ground nodes in our scenario. Each ground node has two radios. We use different channels for communication between mobile node to ground node and ground node to ground node.

The mobile sensor node transmits a packet of 12 B. The sampling rate of the data samples transmitted by the mobile node is five samples per bit. Every bit is made up of complex and real component. Each real and complex component needs 4 B to represent in C++.

$$12 \times 5 \times 8 \times 2 \times 4 = 3840\,\mathrm{B} \tag{4.1}$$

Therefore, ground nodes transmit 3840 B in the ground network for each sample they receive from the mobile node. We have used IEEE 802.11 to communicate between sensor nodes.

According to Yoon, Liu, and Noble [42], random waypoint mobility model does not provide a steady speed throughout the simulations. The speed of the mobile node decreases consistently over the time of the simulation. Thus resulting in the concentration of transmission at the centre of our system area. To overcome this challenge we have concentrated our sensor nodes, which is called the hunting area, at the centre of the system area. We recorded all the points at which mobile nodes transmit over the simulation time. We have plotted all the points of transmission in Figure 4.1, we can see that the number of transmissions grows denser towards the centre of the system area. Hence moving the hunting area in the centre of the system area, marking it by a square coloured in green, helps us to achieve an almost uniform distribution of transmission over the hunting area.

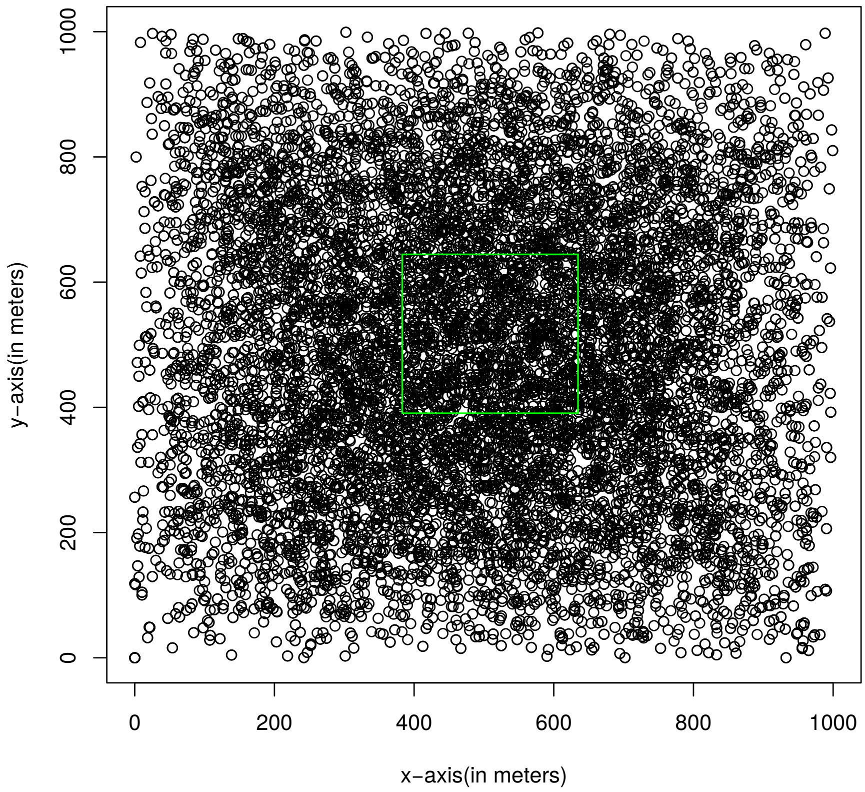



**Figure 4.1** – Mobile node transmission points, according to random waypoint mobility.

In our setup, we have added channel utilization measurement. We record the total time for which radio is either transmitting or receiving, in every 100 ms slot. To validate our implementation we have also developed a mathematical model replicating a simple scenario where every ground node in the network transmits once, which is forwarded to the sink. In the process, we record the ground nodes receiving power more than the sensitivity of the radio, which is −85 dBm, in each 100 ms slot. As ground nodes transmit 3840 B of data in one transmission with bit rate of 24 Mbps, it takes approximately 1.5 ms ( including RTS, CTS and ACK messages by the mac layer of IEEE 802.11), for transmission of a packet to complete. This approximation gives us the time for which ground nodes experience the channel to be busy in 100 ms slot. We have plotted Figure 4.2, ECDF of the channel utilization recorded for the scenario by our simulation model and our mathematical model.

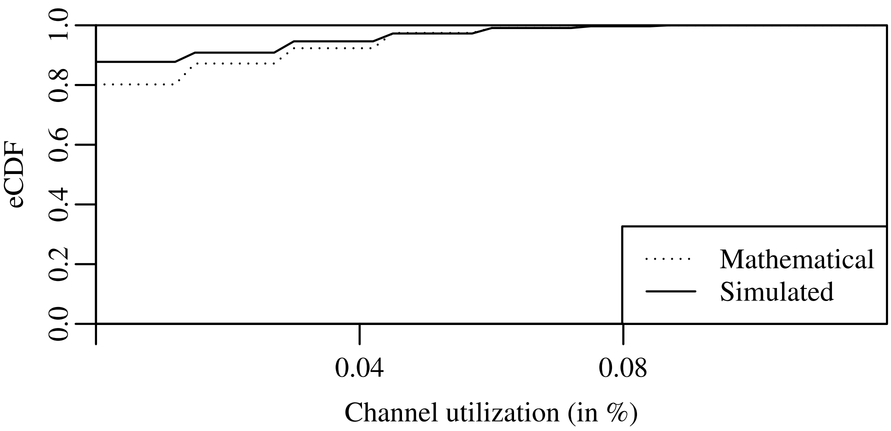
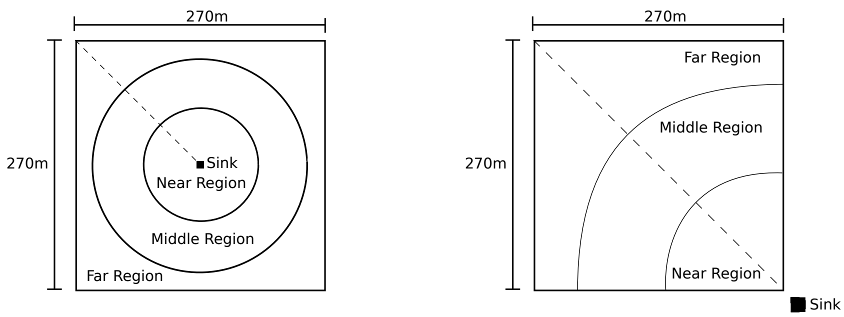



**Figure 4.2** – Channel utilization comparison between simulation and mathematical model.

Data from the ground nodes are forwarded to the sink node. The nodes near to the sink node have a high probability of being involved in a transmission, which is being forwarded to the sink node in comparison to nodes far from the sink. Therefore to study the behaviour of our approaches, relative to the position of transmission made by the mobile node to that of the sink node, we have divided our hunting ground in three regions. We name these regions to be Near, Middle and Far region as seen in Figure 4.3.

In the scenario (Figure 4.3a), with the sink node at the centre of the hunting ground, we calculate the shortest distance between the sink node and one of the vertexes of the hunting area. We draw two circular arcs, such that they divide the line joining the vertex and the centre of the hunting area in three equal parts. The area enclosed by the smallest arc is called the near region, the area between the larger arc and the smaller arc is called the middle region and the area outside the larger arc




**(a)** Sink at center. **(b)** Sink outside.

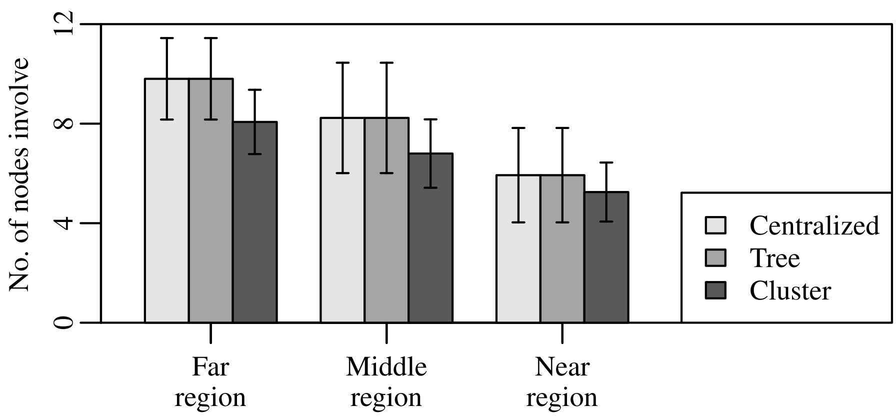
**Figure 4.3** – Hunting area divided into three regions, Far, Middle and Near region, depending on the position of the sink node in the system area.

is called the far region. When the sink is outside the hunting area (Figure 4.3b), we draw a diagonal vertex near to the sink and the opposite vertex. Then we continue to draw arcs in the same way as in the first scenario and mark the regions.

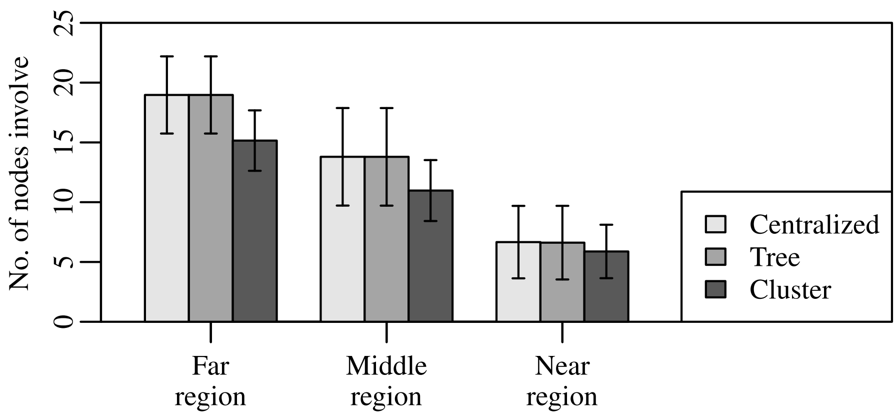
For our further discussion, we will classify the transmissions by the mobile node into these three regions depending on the position of the mobile node while transmission. Consequently discussing its effect on our proposed approaches.

## 4.2 Number of Nodes Involved

Firstly we investigated the number of ground nodes involved in the process of gathering and forwarding the data to the sink node.




**(a)** Sink in centre.




**(b)** Sink outside.

**Figure 4.4** – Number of ground nodes involved for the proposed algorithms to forward the signal successfully to the sink node.

On getting data samples from the mobile node, ground node needs to forward to the sink node. All ground nodes are not one hop away from the sink node in the network discussed in Chapter 3. Ground nodes use other ground nodes as a relay to forward the data to the sink node. Analyzing the ground nodes involved in the process of forwarding the data gives us a fair idea of the total number of ground nodes that directly participate either in processing or forwarding the received mobile node signal for all algorithms and help us estimate the energy load distribution among the nodes.

Data gathering algorithms may forward multiple copies of the samples to the sink node and use a node multiple time as a relay in the process; hence, we have counted each node only once in our study irrespective of the number of times of its involvement.

We have studied this metric for two positions of the sink node as shown in Figure 4.4. On the x-axis, we have three different regions, and on the y-axis, we have the number of nodes involved. The bars indicate the average number of nodes involved and the error bar indicate standard deviation from the mean.

An average number of nodes involved is less in the case of the sink node at centre (Figure 4.4a), compared to when the sink node is outside the network(Figure 4.4b). This effect can be credited entirely to the topology of the network and the position of the sink node.

The mean of the nodes involved reduces as we move from far region to near region, as the hop distance to the sink node reduces.

In both the cases, the nodes involved in case of centralized and tree algorithm is the same in every region. It comes from the fact that in both the algorithms ground nodes in principal forward the data towards the sink, thus taking the same route. However, the cluster algorithm on average involves lesser number of nodes as the ground nodes getting data only take part in cluster formation and only the node selected as CH forwards the data to the sink node.

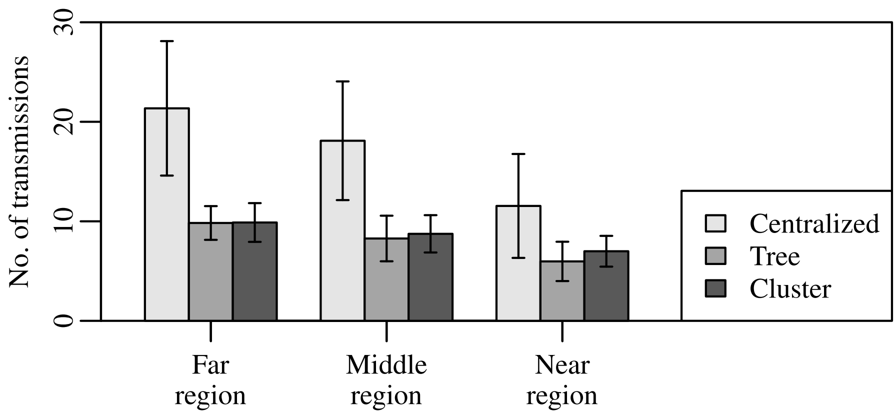
The difference between the means of the other two algorithms to that of cluster algorithm reduces as we move from far region to the near region as the hop distance to sink node is reduced, and the difference in the path taken by the samples to reach the sink node reduces.

The cluster algorithm performs better than the other two algorithms. The number of nodes involved in the process of combining and forwarding the data samples to the sink node is always less.

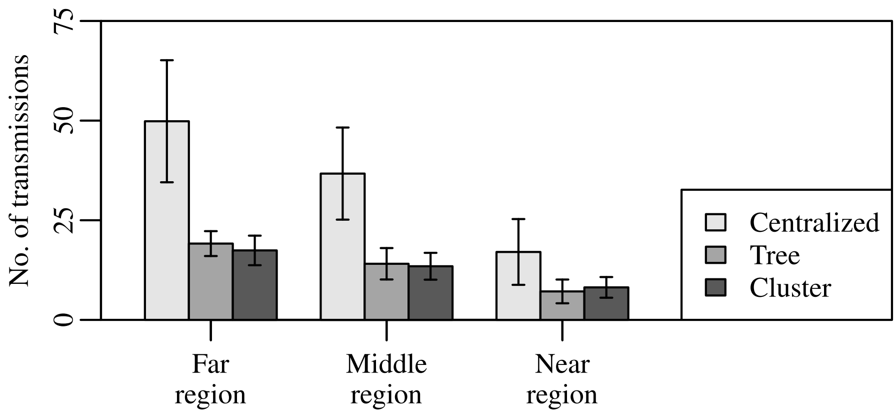
## 4.3 Number of Transmissions

As nodes involved gives us a good measure of the nodes acting as a relay but does not quantify how many transmissions happened in the process of combining and forwarding the data samples.

We record the number of transmissions made in each algorithm shown in Figure 4.5, which includes any message passed to get the ground nodes to co-operate, process or forward the data towards the sink node. Mean of the transmission is indicated by bars in the plot and error bar indicate the standard deviation. We can see that the number of transmissions made to reach the sink node in case of sink outside the hunting area is almost double when compared to the case when the sink node is at the centre of the hunting area.




**(a)** Sink in centre.




**(b)** Sink outside.

**Figure 4.5** – Number of transmissions in the ground network for all the proposed algorithms to forward the signal successfully to the sink node.

The centralized algorithm has the worst performance when compared to other two algorithms, as each ground node forwards its data sample to the sink node without any processing, combining is performed at the sink node. Cluster and tree algorithms take almost the same number of transmission to forward the data sample to the sink. Cluster algorithm marks the cluster head as the aggregator of the copies of the signal into one stream. The diversity combining is used as an aggregation technique; thus only one copy is forwarded towards the sink node. In tree algorithm, the nodes on the way to the sink node wait for a time equal to its calculated backoff and on receiving copies of signal perform diversity combining, and a single stream of data is forwarded. Thus cluster and tree result in almost the same number of transmissions in our study.

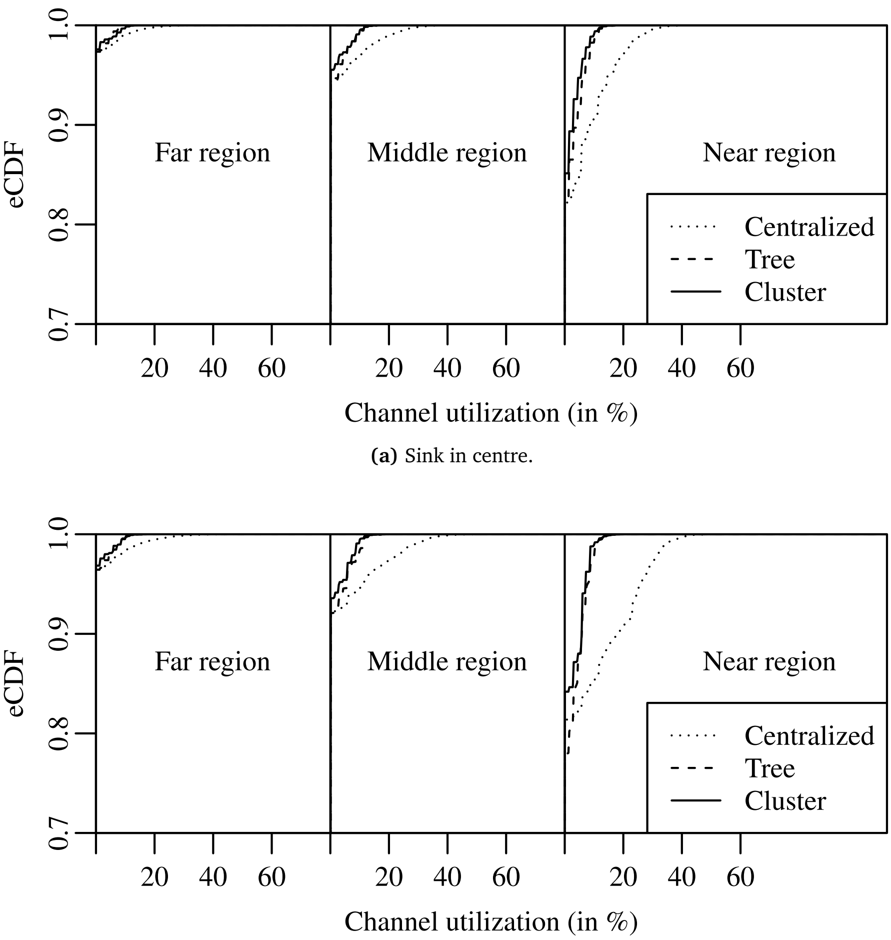
One interesting pattern can be viewed Figures 4.5a and 4.5b, as we move from far region to near region, the gain in cluster algorithm starts to reduce marginally when compared to tree algorithm. Cluster algorithm uses an extra message to form a cluster initially, hence, in the region near to the sink node cluster formation proves to be marginally expensive. Thus we can conclude that our both tree and cluster algorithm perform better than the centralized algorithm irrespective of the region in which transmission happens. However, cluster and tree algorithm performance depends a lot on the position of the transmission. Tree algorithm performs marginally better in cases when the transmission is made near to the sink node, and the cluster algorithm performs better than the tree algorithm as we move away from the sink node.

The tree and cluster algorithms involve about three times and two times fewer transmissions compared to the naïve centralized algorithm when the sink is outside and at the centre respectively.

## 4.4 Channel Utilization

We saw the number of transmissions required varied with different algorithms proposed. Less number transmission means less number of time the channel will get accessed by the ground nodes in the hunting area, thus affecting the time for which ground node sees the channel to be busy. To see the impact of the reduced number of transmission, we recorded channel utilization for both the position of the sink node. We recorded channel utilization every 100 ms, as bat transmits every 100 ms, over the whole simulation time.

We plot the ECDF of channel utilization of the nodes in each region over all the slots during the simulation in Figure 4.6.




**(a)** Sink in centre.



**(b)** Sink outside.

**Figure 4.6** – Channel utilization of ground nodes in different regions.

We can see that more ground node see the channel to be busy in the case when the sink node is outside to when the sink node is at the centre, in every region. It is explained by the fact that we need more number of transmissions to reach the sink node in the case when the sink node is outside to when the sink node is at the centre, in every region.

The fraction of ground node seeing no channel utilization decreases as we move near to the sink node. The ground nodes in the far region are distributed over a large area, thus in case of a transmission, tiny fraction of nodes see the channel to be busy. Middle region area is relatively smaller than far region hence on transmission higher fraction of nodes see the nodes to be busy. At last the near region area is the smallest, therefore on transmission highest fraction of nodes see the channel to be busy.

However, as all the copies converge at the sink node, the nodes nearer to the sink node experiencing non-zero channel utilization see high channel utilization, as they see the channel to be busy on every sample forwarded to the sink node. Hence explaining the trend of increase in channel utilization for the nodes with a decrease in the distance from the sink node.

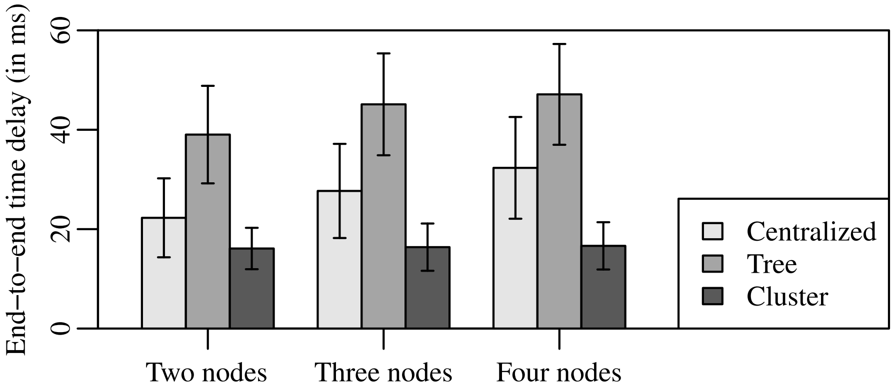
On comparing the algorithms, we see that the clustering algorithm is always better or equivalent to the tree algorithm but much better compared to the centralized algorithm in all cases. Centralize algorithm makes the highest number of transmission as the ground node on receiving the sample from the mobile node, send each copy of the sample to the sink node through different routes; thus more nodes see the channel to be busy due to the high number of transmission made in the network.

## 4.5 Time Delay

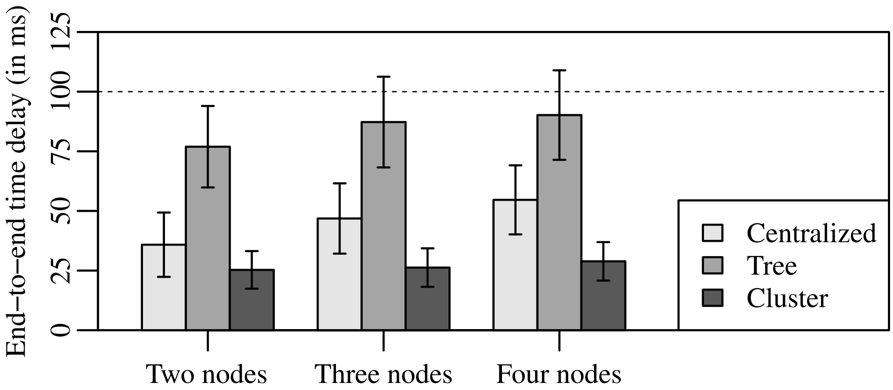
We have analyzed the end-to-end delay of all the algorithms. End-to-end delay is the time taken by the data sample transmitted by the mobile node to reach the sink node. Time taken by algorithms to forward the data to the sink node plays a vital role in time-sensitive applications. In Figures 4.7 and 4.8, we plot the mean end-to-end time delay of the considered algorithms. The bar represents mean of time taken by algorithms and error bars show the standard deviation.

Nabeel, Bloessl, and Dressler [5] studied the effect of employing diversity combining techniques two and three branches on the PDR(%). They showed that PDR improved with an increase in the number of branches. Hence, we measure the time taken by our algorithms to gather and forward the samples to the sink node with at least two, three and four branches; i.e., samples from two, three and four ground nodes.

In Figure 4.7 we can observe an interesting behaviour, as the number of branches increases the cluster and tree algorithm, time delay increases but the time delay of the cluster algorithm remains almost constant. In cluster algorithm, we can divide the time consumed into time required for three steps, i.e., CH selection, data gathering from CM and forwarding data to the sink node. With an increase in the desired number of branches we see a minimal increase in time in the data gathering phase, hence, overall time delay only increases slightly. However, in the case of centralize algorithm, every ground node sends its data independently to the sink node and combining is employed at the sink node. Thus an increase in the number of branches results in a significant increase in the total time delay. The tree algorithm inherently depends on its backoff time calculated at respective levels to gather data, to gather copies we wait at level until backoff expires, contributing to higher waiting time than the other two algorithms. Once the desired number of sample copies is combined, data gets forwarded to the sink node ignoring delay at the further levels. Hence with an increase in the desired number of branches, to gather enough copies algorithm needs to wait at more levels, resulting in even higher time delay.




**(a)** Sink in centre.

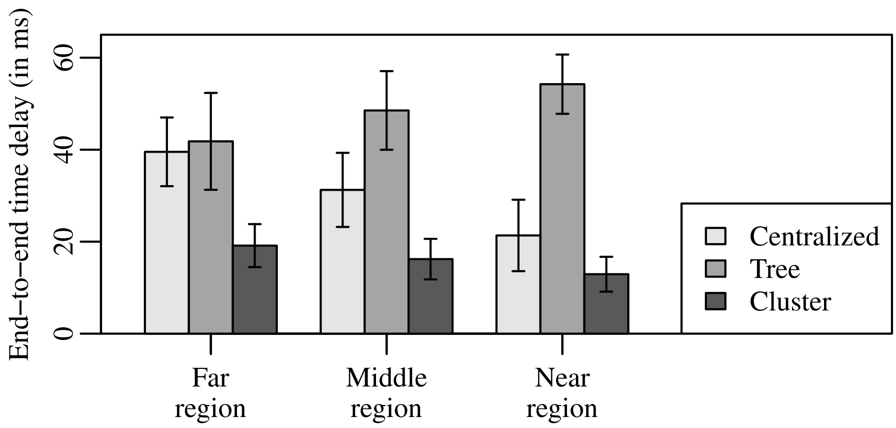



**(b)** Sink outside.

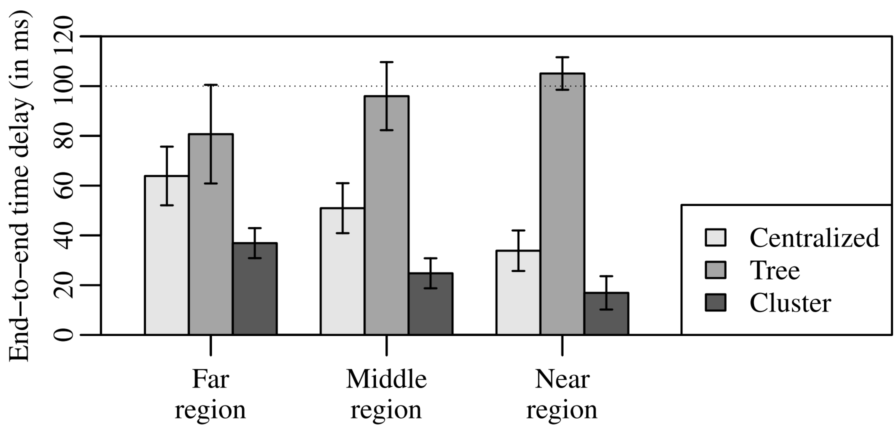
**Figure 4.7** – End-to-end delay for all proposed algorithms to combine at-least two, three and four data sample copies.

In the Figure 4.7, we have taken the mean of the end-to-end time delay of the transmission all over the hunting area. To further understand the behaviour of end-to-end time delay w.r.t., the position of the transmission in the hunting area, we conducted the study to record the end-to-end delay for all the three regions setting the number of branches to at least four.

We present the result in Figure 4.8. We can see two important trends in the plot, as we move nearer to the sink node, time delay reduces with centralized and cluster algorithm, however, tree algorithm shows just the opposite trend. This effect can again be explained by another effect, as we move near to the sink node centralized and cluster algorithm need lesser number of transmission to forward its data to the sink node, hence taking lesser time. However, increase in the time delay in tree algorithm as we move near to the sink node cannot be credited to the number of transmissions. This effect stems from the fact that the tree algorithm relies on *level*-based backoff calculation. As we move near to the sink node the level value increases, hence resulting in high backoff values, consequently resulting in a higher time delay.




**(a)** Sink in centre.




**(b)** Sink outside.

**Figure 4.8** – End-to-end delay for all proposed algorithms to combine at-least four data sample copies for bat transmission in different regions.

In cluster algorithm irrespective of the region time required to form a cluster and gather the data does not change a lot, the time to forward the combined data varies as a number of transmissions required changes. As forwarding of combined data is one of the factors contributing to the time delay, a decrease in the number of transmissions does not impact the time delay much, instead decrease in the time delay is gradual. However, centralized algorithm sends all the copies to the sink node, hence when we move near to the sink node we see a reduction in the number of transmissions required to make all the four copies available at the sink node. Thus we see considerably high time delay for transmissions in the far region, and it reduces proportionally to the reduction in the number of transmissions over the regions.

In the case of sink outside the hunting area, we observe worst-case time delay for tree algorithm in the near region. The time delay crosses 100 ms, marked by a dotted line. We have discussed every bat re-transmit after every 100 ms, interference between two transmissions from the same mobile node may result in loss of data samples.

It is clear from the above discussion that the cluster algorithm results in the least time delay out of all the algorithms and the time delay does not change drastically depending on the position of the transmission or the number of branches used for combining.

## 4.6 Energy Consumption

Finally, we studied the energy consumption of the ground nodes in the hunting area, as a metric for the energy load distribution within the hunting area in the network. As we have discussed, ground nodes use the IEEE 802.11 protocol to communicate with each other. Halperin et al. [45] studied the effect of various parameters like transmit power, antenna, data rates on the energy consumption of an 802.11n NIC. The findings retreated the findings in [46], that transmit power control provides little gain as it is a small fraction of total power consumption. The findings also brought out the effect of data rate on power consumption.

For our study, we consider 50nJ per bit of energy consumption and bitrate is 24 Mbps. We further calculated power consumption using the formulae provide in [45].

$$power\ consumption = per - bit\ energy \times bitrate \tag{4.2}$$

Hence on calculating, we get the power consumption to be 1.2 W for each transmission and reception, we have not considered the energy consumption in the ideal state by the IEEE 802.11 NIC.

Considering mobile node transmit after every 100 ms throughout the simulation in total mobile node makes approximately 23000 transmissions. We have recorded the energy consumption by the ground nodes for each transmission and reception, further, we added the energy consumption of the nodes over the complete period of simulation and runs.

Further, we plot the energy consumption of the ground nodes in Figures 4.9 to 4.11 employing centralized, cluster and tree algorithm respectively. The plots represent the state of the network after 23000 mobile node transmissions.

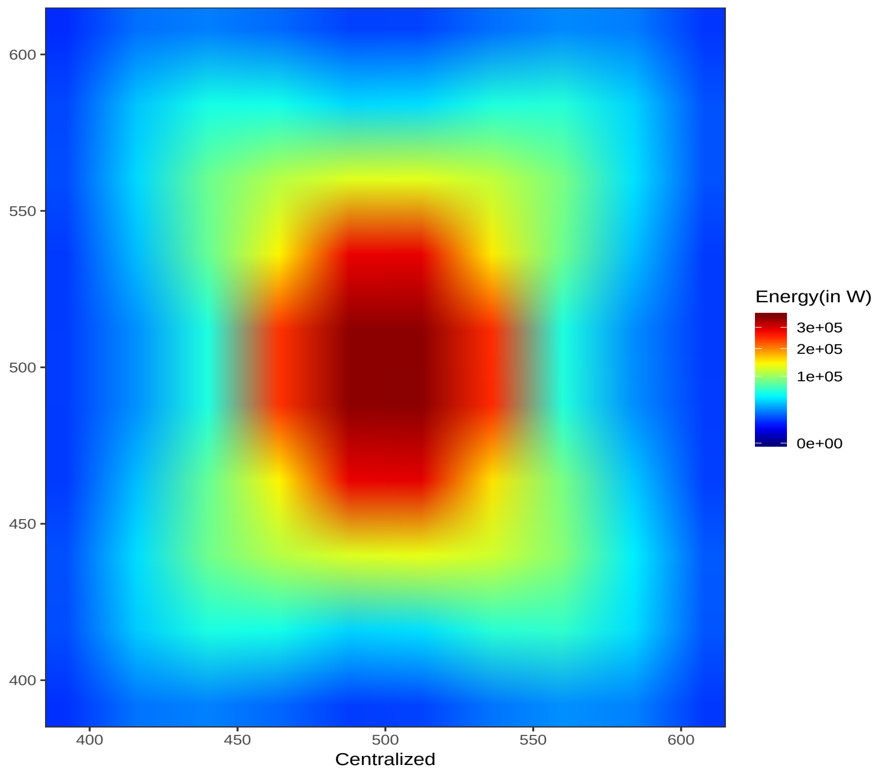



**Figure 4.9** – Energy consumption in ground nodes employing centralized algorithm.

The x-axis and y-axis represent the actual position of the hunting area in the system area; hence, plots start at 365 on x-axis and y-axis. The legend represents the energy consumption value of the ground nodes. The legend scale have been converted to square root for better readability of plots.

Analyzing the plots, we observe as we move near to the sink node energy consumption in the ground nodes increase. As data generated in the ground network funnels down to the sink node, the ground nodes near the sink node are involved in more number of reception and transmission of data samples. Hence resulting in higher energy consumption when compared to ground nodes in other parts of the network.

One more interesting pattern we witness is that the energy consumption of nodes near the sink node is approximately three times higher in case of the centralized algorithm when compared to that of cluster and tree algorithm. We do know that the centralized algorithm uses the sink node as an aggregator; thus this very fact also explains this effect. As all the data samples received by ground nodes in the network is forwarded to the sink node, ground nodes perform more transmissions when compared to the cluster and tree algorithm where we perform in-network aggregation.

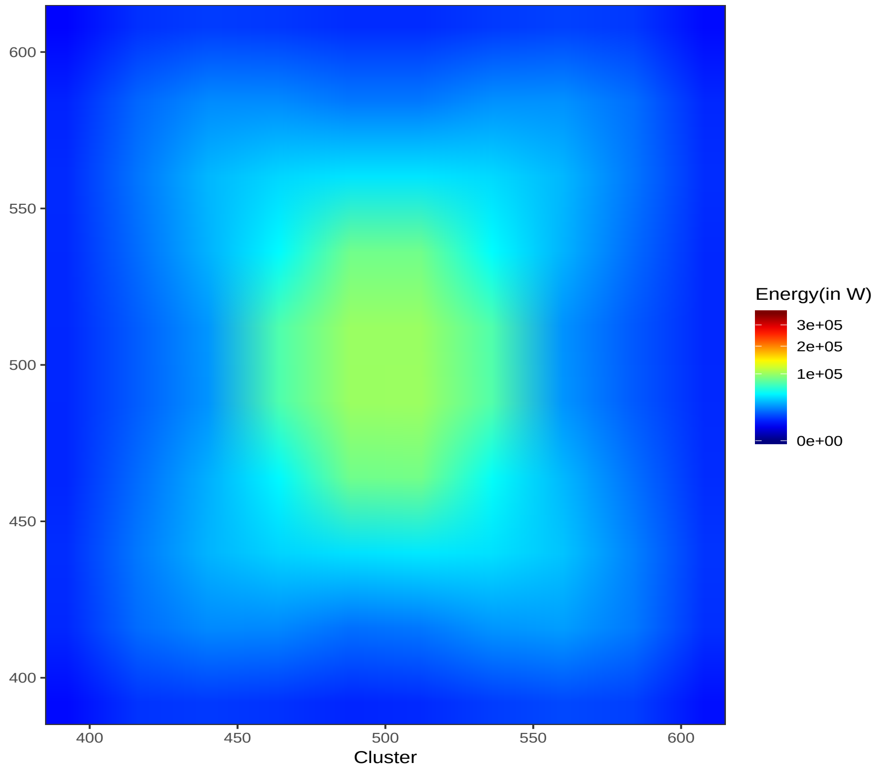



**Figure 4.10** – Energy consumption in ground nodes employing cluster algorithm.

The cluster and tree algorithm has almost the same energy consumption, as we have seen earlier that the number of transmissions in both the algorithms is nearly the same. The interesting point to note here is that the cluster algorithm has better energy distribution and than the tree algorithm. As the once cluster is formed only one stream of data is forwarded towards the sink node whereas in case of tree algorithm aggregation happens on the way to the sink node. Thus resulting in lesser energy consumption on the static routes in case of cluster algorithm than the tree algorithm.

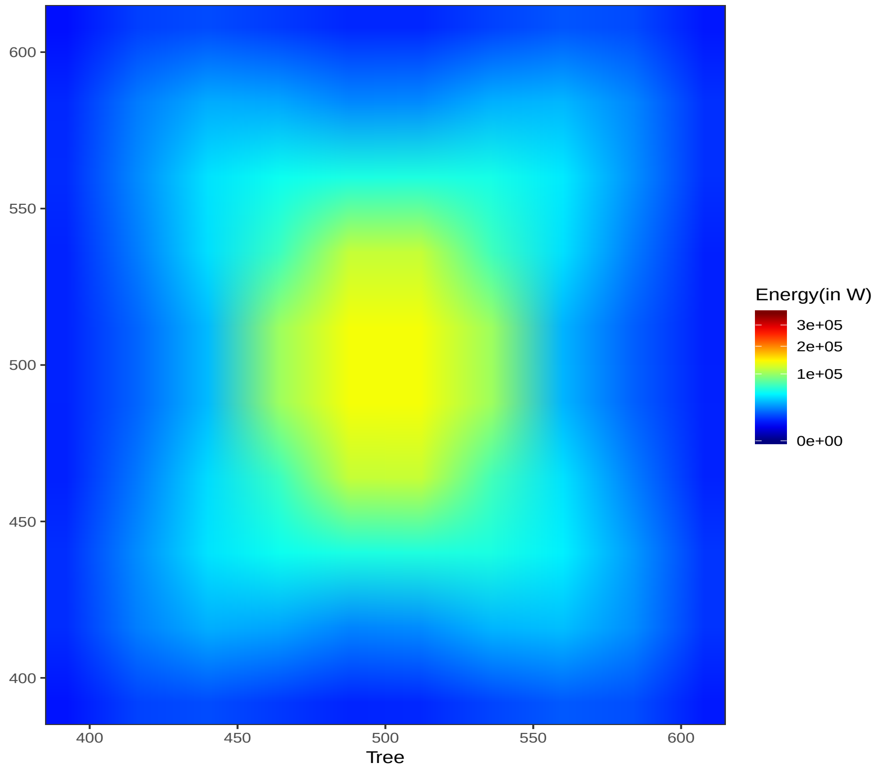



**Figure 4.11** – Energy consumption in ground nodes employing tree algorithm.

We can clearly see that centralized algorithm consumes more energy than that of the cluster and tree algorithms. One more interesting pattern to be observed is the increase in energy consumption as we move near to the sink node. As all the data finally funnels down to the sink node in the WSN, the nodes near to the sink node act as a relay to most of the transmissions or receive to most transmissions around the sink node, resulting in high energy consumption.

The cluster and tree algorithm has almost the same energy consumption, as we have seen earlier that the number of transmissions in both the algorithms is nearly the same. The interesting point to note here is that the cluster algorithm has better energy distribution and than the tree algorithm. The energy consumption in case of cluster algorithm is lesser than the tree algorithms on the static routes to the sink node and around the sink node. As the once cluster is formed only one stream of data is forwarded towards the sink node whereas in case of tree algorithm aggregation happens on the way.

In the end, we summarize the algorithm proving to be the most efficient in respective metric in Table 4.3.

| Metric | Algorithm |
|---|---|
| No. of nodes involve | Cluster |
| No. of transmissions | Cluster & Tree |
| Channel Utilization | Cluster & Tree |
| End-to-end delay | Cluster |
| Energy | Cluster |

**Table 4.3** – Summary of evaluation

Cluster algorithm proves to be the best algorithm considering all the metrics. Tree algorithm shows the same effectiveness considering the number of transmissions, channel utilization and energy footprint. However, it performs worse when we consider end-to-end delay.

# Chapter 5

# Conclusion

In this thesis, we investigated different techniques to efficiently gather data in order to perform diversity combining in distributed sensor networks. We developed three techniques, naïve centralized (which acts as a baseline), cluster-based and, tree-based approaches to solve this problem.

On comparing the three algorithms, we showed that the cluster and tree algorithms perform way better than the centralized algorithm. Cluster and tree algorithm takes two to three times fewer transmissions to complete the process of gathering, combining and forwarding the data finally to the sink node.

In general, the cluster algorithm takes marginally less number of transmissions than that of the tree algorithm. However, as the mobile node transmits near to the sink node overhead to form the cluster outweighs the gain achieved by the cluster algorithm, thus tree performing marginally better in such situations. Both the algorithms therefore also show the same channel utilization pattern in the network.

Overall, both the cluster and tree algorithms perform the same on parameters like number of transmissions, channel utilization. However, tree algorithm performs worse considering the end-to-end time delay, and, cluster algorithm achieves the best end-to-end time delay.

# List of Abbreviations

| | |
|---|---|
| **AF** | Amplify-and-Forward |
| **ANCH** | Avoid Near Cluster Heads |
| **AWGN** | Additive White Gaussian Noise |
| **BER** | bit-error rate |
| **CH** | Cluster Head |
| **CM** | Cluster Member |
| **DF** | Decode-and-Forward |
| **DWEHC** | Distributed Weight-Based Energy-Efficient Hierarchical Clustering |
| **EGC** | Equal Gain Combining |
| **FEC** | Forward Error Correction |
| **FSPL** | Free Space Path Loss |
| **LOS** | line-of-sight |
| **MOCA** | Multi-hop Overlapping Clustering Algorithm |
| **MRC** | Maximal-Ratio Combining |
| **PDR** | Packet Delivery Rate |
| **POO** | Point-of-Origin |
| **RCC** | Random competition based clustering |
| **SNR** | Signal to Noise Ratio |
| **WSN** | Wireless Sensor Network |

# List of Figures

# List of Tables

# Bibliography


[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, Atlanta, Georgia, USA: ACM, 2002, pp. 88–97.

[2] C. Rutz, Z. T. Burns, R. James, S. M. Ismar, J. Burt, B. Otis, J. Bowen, and J. J. S. Clair, "Automated mapping of social networks in wild birds," *Current Biology*, vol. 22, no. 17, R669–R671, 2012.

[3] F. Dressler, B. Bloessl, M. Hierold, C.-Y. Hsieh, T. Nowak, R. Weigel, and A. Koelpin, "Protocol Design for Ultra-Low Power Wake-Up Systems for Tracking Bats in the Wild," in *IEEE International Conference on Communications (ICC 2015)*, London, UK: IEEE, Jun. 2015, pp. 6345–6350.

[4] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-Cell MIMO Cooperative Networks: A New Look at Interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[5] M. Nabeel, B. Bloessl, and F. Dressler, "Efficient Receive Diversity in Distributed Sensor Networks using Selective Sample Forwarding," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 336–345, Jun. 2018.

[6] F. Kirsten, D. Öhmann, M. Simsek, and G. P. Fettweis, "On the utility of macro- and microdiversity for achieving high availability in wireless networks," in *26th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2015)*, Hong Kong, China: IEEE, Aug. 2015, pp. 1723–1728.

[7] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, Apr. 2007.

[8] P. Jesus, C. Baquero, and P. S. Almeida, "A Survey of Distributed Data Aggregation Algorithms," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 381–404, Firstquarter 2015.

[9] Y. Sang, H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong, "Secure Data Aggregation in Wireless Sensor Networks: A Survey," in *2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, Taipei, Taiwan: IEEE, Dec. 2006, pp. 315–320.

[10] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Elsevier Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[12] S. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[13] D. Brennan, "Linear Diversity Combining Techniques," *Proceedings of the IRE*, vol. 47, no. 6, pp. 1075–1102, Jun. 1959.

[14] T. Eng, N. Kong, and L. B. Milstein, "Comparison of diversity combining techniques for Rayleigh-fading channels," *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1117–1129, Sep. 1996.

[15] V. A. Aalo, "Performance of maximal-ratio diversity systems in a correlated Nakagami-fading environment," *IEEE Transactions on Communications*, vol. 43, no. 8, pp. 2360–2369, Aug. 1995.

[16] Y. Tang and M. C. Valenti, "Coded transmit macrodiversity: block space-time codes over distributed antennas," in *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*, vol. 2, Rhodes, Greece: IEEE, May 2001, pp. 1435–1438.

[17] A. Bletsas, H. Shin, and M. Win, "Cooperative Communications with Outage-Optimal Opportunistic Relaying," *IEEE Transactions on Wireless Communications*, vol. 6, no. 9, pp. 3450–3460, Sep. 2007.

[18] J. Laneman, D. Tse, and G. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[19] F. Dressler, S. Ripperger, M. Hierold, T. Nowak, C. Eibel, B. Cassens, F. Mayer, K. Meyer-Wegener, and A. Koelpin, "From Radio Telemetry to Ultra-Low-Power Sensor Networks: Tracking Bats in the Wild," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 129–135, Jan. 2016.

[20] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance Measurements of Motes Sensor Networks," *7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '04, pp. 174–181, 2004.

[21] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *33rd Hawaii International Conference on System Sciences*, 2000.

[22] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[23] J. Leu, T. Chiang, M. Yu, and K. Su, "Energy Efficient Clustering Scheme for Prolonging the Lifetime of Wireless Sensor Network With Isolated Nodes," *IEEE Communications Letters*, vol. 19, no. 2, pp. 259–262, Feb. 2015.

[24] P. Nayak and A. Devulapalli, "A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 137–144, Jan. 2016.

[25] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *MILCOM 2002. Proceedings*, vol. 2, Anaheim, CA, USA: IEEE, Oct. 2002, pp. 838–843.

[26] P. Ding, J. Holliday, and A. Celik, "Distributed Energy-efficient Hierarchical Clustering for Wireless Sensor Networks," in *Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS'05, Marina del Rey, CA: Springer-Verlag, 2005, pp. 322–339.

[27] A. Youssef, M. Younis, M. Youssef, and A. Agrawala, "WSN16-5: Distributed Formation of Overlapping Multi-hop Clusters in Wireless Sensor Networks," in *IEEE Globecom 2006*, San Francisco, CA, USA: IEEE, Nov. 2006, pp. 1–6.

[28] J. Yu, Y. Qi, G. Wang, and X. Gu, "A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution," *AEU-International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 54–61, 2012.

[29] M. M. Zanjireh, A. Shahrabi, and H. Larijani, "ANCH: A new clustering algorithm for wireless sensor networks," in *27th International Conference on Advanced Information Networking and Applications Workshops*, IEEE, Barcelona, Spain, Mar. 2013, pp. 450–455.

[30] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[31] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, Sep. 2002.

[32] M. Ding, X. Cheng, and G. Xue, "Aggregation Tree Construction in Sensor Networks," in *58th IEEE Vehicular Technology Conference (VTC2003-Fall)*, vol. 4, Orlando, FL: IEEE, Oct. 2003, pp. 2168–2172.

[33] T. Kuo and M. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms," in *31st IEEE Conference on Computer Communications (INFOCOM 2012)*, Orlando, FL: IEEE, Mar. 2012, pp. 2591–2595.

[34] H. Lin and W. Chen, "An Approximation Algorithm for the Maximum-Lifetime Data Aggregation Tree Problem in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3787–3798, Jun. 2017.

[35] F. Dressler, M. Mutschlechner, B. Li, R. Kapitza, S. Ripperger, C. Eibel, B. Herzog, T. Hönig, and W. Schröder-Preikschat, "Monitoring Bats in the Wild: On Using Erasure Codes for Energy-Efficient Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 12, no. 1, Feb. 2016.

[36] M. Nabeel, B. Bloessl, and F. Dressler, "Low-Complexity Soft-Bit Diversity Combining for Ultra-Low Power Wildlife Monitoring," in *IEEE Wireless Communications and Networking Conference (WCNC 2017)*, San Francisco, CA: IEEE, Mar. 2017.

[37] A. Varga, "The OMNet++ Discrete Event Simulation System," in *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, Jun. 2001.

[38] X. Xian, W. Shi, and H. Huang, "Comparison of OMNET++ and other simulator for WSN simulation," in *3rd IEEE Conference on Industrial Electronics and Applications*, IEEE, Singapore, 2008, pp. 1439–1443.

[39] C. Wei, J. Yang, Y. Gao, and Z. Zhang, "Cluster-based routing protocols in wireless sensor networks: A survey," in *International Conference on Computer Science and Network Technology*, IEEE, vol. 3, Harbin, China, Dec. 2011, pp. 1659–1663.

[40] M. M. Zanjireh and H. Larijani, "A Survey on Centralised and Distributed Clustering Routing Algorithms for WSNs," in *81rd IEEE Vehicular Technology Conference (VTC2015-Spring)*, Glasgow, UK: IEEE, May 2015.

[41] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., vol. 353, Kluwer Academic Publishers, 1996, pp. 152–181.

[42] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," in *22nd IEEE Conference on Computer Communications (INFOCOM 2003)*, San Francisco, CA: IEEE, Mar. 2003, pp. 1312–1321.

[43] D. L. Mills, "Internet Tme Synchronization: the Network Time Protocol," *IEEE/ACM Transactions on Networking (TON)*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[44] M. Nabeel, B. Bloessl, and F. Dressler, “On Using BOC Modulation in Ultra-Low Power Sensor Networks for Wildlife Tracking,” in *IEEE Wireless Communications and Networking Conference (WCNC 2016)*, Doha, Qatar: IEEE, Apr. 2016, pp. 848–853.

[45] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, “Demystifying 802.11N Power Consumption,” in *Proceedings of the 2010 International Conference on Power Aware Computing and Systems*, ser. HotPower’10, Vancouver, BC, Canada: USENIX Association, 2010.

[46] F. I. Di Piazza, S. Mangione, and I. Tinnirello, “On the Effects of Transmit Power Control on the Energy Consumption of WiFi Network Cards,” in *Quality of Service in Heterogeneous Networks*, Berlin: Springer, 2009, pp. 463–475.