**Abstract**

A wireless sensor network (WSN) is an aggregation of cooperative sensor nodes acting with each other in different environments to monitor events of interest. Energy efficiency has become a main challenge in WSNs and their applications, and thus energy efficiency of localisation is one of the most active research areas in WSNs. Localisation generally refers to the process of locating the position(s) of one or more node(s) in a network. The challenge of localisation lies in efficiently providing acceptable accuracy while conforming to the many constraints of WSNs, such as the limited lifetime of sensor nodes. This thesis develops and evaluates an energy savings-based localisation algorithm in WSNs. Savings in nodes’ energy consumption lead to enhancing network lifetime, and clustering techniques have been presented in the literature as an efficient technique for producing such savings.

The proposed approach is as an integration technique that supports both accurate localisation and efficient energy consumption in the network. It is mainly based on Alwadha algorithm, which stands for ‘an efficient localisation algorithm for wireless ad hoc wireless sensor networks with high accuracy’, and is an efficient algorithm used to accurately locate the position of sensor nodes. It does not rely on the use of a high number of references to enhance the accuracy of estimation; rather, it relies on the use of a smart reference selection method.

The proposed algorithm is implemented using a WSN simulator, Castalia 3.2/OMNet++. The experimental results indicate that the proposed integrated technique is more effective in terms of reducing overall network energy consumption while not sacrificing the accuracy of node localisation. Indeed, a comparison is done between the original and the new integrated algorithm to show the significance of the proposed modification. The new approach saves the most energy when there is a traffic load with a large network size or in grid deployment because of the new feature of selecting cluster head nodes. Another part of the results is concentrated on noticing the effect of using different MAC layer protocols on the localisation algorithm. Two MAC protocols, TMac and TunableMac, are used. In the first protocol, sensor nodes are turned on and off at synchronised times, while the second is connection based and uses a code division multiple access (CDMA) mechanism to mitigate packet collisions.

**1 Introduction**

* 1. **Overview**

A wireless sensor network (WSN) is a network of autonomous devices called sensors that is used to monitor physical or environmental situations, such as temperature, motion, pressure, vibration, or pollutants, at different locations. Each node in a sensor network is typically armed with a radio transceiver or other wireless communication device, a small micro controller, and an energy source, commonly a battery. Size and cost limitations on sensor nodes result in identical limitations on resources such as energy, memory, computational speed, and network bandwidth. There is a major need for automated network self-configuration and observation of WSNs, and this requirement will only in increase as networks grow bigger.

In this dissertation, we study a self-configuration method of estimating the location of sensor nodes. Sensors are used to indicate different devices connected throughout the network while node locations indicate the coordinates of each node in the network.

Most applications of WSNs require knowledge of the position of the sensing information. This process of defining the position or location is called localisation. According to different studies, the global positioning system (GPS) is suitable for solving the problem of localisation in outdoor environments, but only for PC-class nodes. For WSNs the situation is different since the network is large and consists of small, cheap, low power devices. These devices have practical limitations such as size, cost, and energy constraints that prevent the application of GPS on all nodes of the network. The use of GPS in WSNs is difficult because:

* GPS cannot be executed in dense forests or mountain settings because these features block the line-of sight from GPS satellites.
* The battery lifetime of the sensor nodes is decreased due to the impact of the energy consumption of GPS, which decreases the efficient lifetime of the entire network.
* Due to the large number of nodes in a WSN, the manufacturing cost of GPS is an important factor.
* Sensor nodes must be small, but the size of GPS and its antenna would maximise the sensor node shape.

For these reasons an alternate GPS solution is required that is cheaper, quickly deployable, and can run in varied environments. There are various challenges in designing effective and strong sensor localisation algorithms for real sensor network applications, which may be summarised as follows:

1. A large number of sensors is commonly utilised, and they are randomly deployed through a specific area. The researchers hope to obtain good position estimates while keeping the hardware layout of the sensor devices simple and inexpensive.
2. In many circumstances, it is impossible to fetch a large number of known nodes deployed uniformly through the area to aid in location estimation of unknown nodes. Thus, there is a need to design a sensor localisation method capable of producing accurate localisation estimates with as few known nodes as possible.
3. In many situations, sensors may be deployed in an area with anisotropic vegetation and topography conditions. Thus, the sensors may have various radio ranges, and using a uniform radio range computation will result in serious errors during sensor localisation—and such errors will compound throughout the sensors in the WSN.
4. Finally, most current sensor localisation research attempts to provide accurate location estimation for sensor networks. WSNs have limited energy resources, but sensor localisation generally requires energy consuming calculations and communication. Thus, it is desirable to decrease the energy costs related to sensor localisation with improved localisation methods that are able to locate sensors on demand. Many applications and operations in sensor networks only require location information for some sensors, which enables on-demand sensor localisation. Some sensor networks are mobile or deployed in a dynamic environment, for example, in a lake or sea to observe fish actions or water pollution. Such an environment leads to slowly and constantly drifting sensors, meaning that the estimated locations of sensors are quickly invalidated. In this case, it is hard to find all nodes in the sensor network. A better alternative is to find the position of the ‘right’ sensors at the ‘right’ time.

In general, energy consumption is a key factor in WSN localisation. Considerable research has been conducted to determine a good balance between energy consumption and object precision. Clustering is a standard approach that is used to achieve efficient energy and scalable performance in WSNs. Clustering allows the distribution of control over the network and, hence, enables locality of communication. Clustering nodes into groups saves energy and decreases network disputes because nodes communicate their data over shorter distances to their cluster heads. The cluster heads then forward the aggregated information to the base station. Thus, only the cluster heads transfer far distances to the base station.

Medium access control (MAC) plays a pivotal role in power savings in WSNs. Several MAC layer protocols have been proposed and executed in real applications in WSNs. Network lifetime, energy consumption, latency, and reliable transmission of packets are some of the main issues in WSNs. Moreover, battery, memory, and processing capabilities are limited, and because of this, researchers need to search for a highly efficient MAC layer transmission protocol.

Considering the challenges of sensor localisation, this thesis aims to study the localisation problem, how to save energy in WSNs, and examine the impact of MAC protocols on localisation. We studied the practical problem to discover the trade-off between accuracy and energy costs while achieving an excellent sensor network localisation algorithm.

**1.2 Problem Statement**

WSNs have small nodes that autonomously sense, calculate, and communicate data between themselves. Energy efficiency is one of the key design issues in WSNs as energy consumption is directly related to network lifetime. Due to the harsh environments in which these networks are used, it is not possible to exchange batteries on a regular basis. WSNs are employed in a wide range of applications such as environmental monitoring, object detection, battlefield and military, target tracking, and civil aviation.

In many WSN applications, including observation and object tracking, the collected data is useless without knowledge of the positions of the sensor nodes. Sensor localisation in ad hoc WSNs thus aims to determine the positions of all sensors in the network. Since the localisation algorithm is executed by every single node, the solution has to be comparatively simple and demand limited resources (in regard to calculations, memory, and communication overhead). The performance of localisation algorithms depends on critical network sensor parameters, such as the density of nodes, the number of known and unknown nodes, deployment type, and network size. It is important that the solution provides adequate performance over a range of reasonable parameter values. Further, it should reduce the energy consumption of the WSN.

**1.3 Thesis Statement**

The objective of this thesis is to develop a technique for balancing the energy consumption of localisation in WSNs. This will be done by integrating the concepts of localisation and clustering into a single algorithm called the Alwadha position clustering algorithm. The new generated algorithm should achieve good accuracy of position estimation while reducing energy consumption. The inputs and expected output of the thesis are summarised below:

* Input
  + Total Nodes: The total number of nodes in the network
  + Unknown Node: A node with an unknown location
  + Known Node: A node with a known location
* Output
  + Locations: Estimated locations for sensors

This objective will be achieved after surveying and studying current approaches for localisation and clustering in WSNs. Moreover, the developed technique will be evaluated to ensure its applicability and measure its performance.

**1.4 Thesis Contributions**

In this thesis, we propose an algorithm for localisation to save energy in WSNs. The following summarise the thesis contributions:

1. Study different existing beacon-based distributed localisation algorithms and select the Alwadha algorithm as a performance reference.
2. Propose a new energy-based localisation algorithm called the Alwadha position clustering algorithm.
3. Implement both the reference algorithm and the proposed algorithm using the Castalia 3.2/Omnet++ simulation tool.
4. Register and analyse the effect of using different MAC protocols on the performance of the reference algorithm.
5. Conduct a comparative analysis of the new model and the original localisation algorithm in terms of accuracy and energy consumption.

**1.5 Thesis Outline**

The basic sensor localisation concept and current methods are described in the first chapter. Chapter 2 presents background information on WSNs and the state of the art of different technical solutions as well as the most promising algorithms. Chapter 3 explains the proposed algorithm and its design. Chapter 4 presents the MAC layer and the effects of MAC protocols on the performance of the Alwadha algorithm with the implementation of two of the protocols: TMac and TunableMac. Chapter 5 provides implementation details of the proposed technique. Potential simulators for implementing this technique are surveyed, and the selection of the OMNeT++ simulator is explained. Then, details of the simulation environment and its modules are provided along with information about the experiments that were used to test the proposed technique and the results obtained. Chapter 6 summarises the conclusions of the conducted research and presents directions for future work. In the simulator appendix ‘Chapter 8’, details for installing the OMNeT++ simulator and its extensions are presented.

**1.6 Summary**

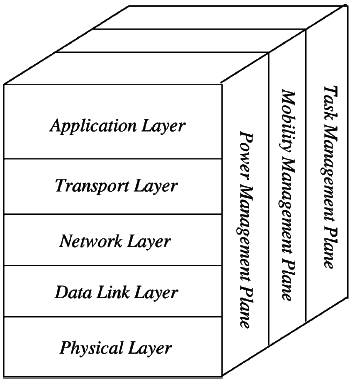
In summary, this chapter is an introduction to the thesis. First, an overview is given of WSNs and their applications as well as the architecture of localisation of wireless sensor nodes. Then, the integration between localisation and the clustering unit is introduced with a brief overview of different known approaches. Finally, the problem statement, thesis statement, thesis objectives, contributions, and the outline of this thesis are presented.

**2 Background and Related Research**

**2.1 Background of WSNs**

**2.1.1 Sensor Network Architecture**

Most common architecture for WSNs follows the OSI model. A sensor network needs five layers: application layer, transport layer, network layer, data link layer, and physical layer. Added to these five layers are the three cross layer planes, as shown in Fig. 2.1 [30].



*Figure 2.1*. WSN architecture.

**2.1.1.1 Cross Layers**

The three cross layers are the power management plane, mobility management plane, and task management plane. These layers are applied to run the network and enable the sensors to work together to improve the overall efficiency of the WSN.

**2.1.1.2 WSN OSI Layers**

*Application Layer:* Responsible for traffic department and supply software for different applications that interpret the data in an understandable form or transmit queries to obtain specific information. Sensor networks are deployed in different applications in various fields, for instance, medical, marine, environmental, and agricultural fields as well as in the military. In this thesis I focused on this layer when I added the new cluster head selection algorithm have been proposed Alwadha position clustering algorithm (explained in Chapter 3).

*Transport layer:* The function of this layer is to supply reliability and congestion avoidance, and many of the protocols designed to provide this function are either used on the upstream (user to sink) or downstream (sink to user). These protocols employ different techniques for loss detection ((ACK, NACK, and sequence number)) and loss recovery ((end to end or hop by hop)). This layer is specifically necessary when a system is orderly to access other networks.

*Network Layer:* The main task of this layer is routing. This layer faces a number of challenges depending on the application, but the main challenges are in power saving, limited memory, and buffers. Sensors do not have a global ID and must be self-organised. This is unlike computer networks with an IP address and central device for controlling. The basic concept of the routing protocol is to know a reliable path and redundant paths according to a specific scale called metric, which differs from protocol to protocol. An example of a routing layer is the low energy adaptive clustering hierarchy (LEACH) [30].

*Data Link Layer*: Responsible for multiplexing data streams, data frame detection, MAC, and error control. The MAC layer is responsible for channel access policies, scheduling, buffer management, and error control. In WSNs, we need a MAC protocol to examine energy efficiency, reliability, low access delay, and high throughput [30]. In this thesis I also focused on this layer when I presented two MAC layer protocols, TMac and TunableMac (explained in Chapter 5).

*Physical Layer:* May supply an interface to send a stream of bits over a physical medium. It is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption [30].

**2.2 Related Research**

Since investigation of the WSN localisation problem has been presented in this thesis, a literature survey on localisation is important. In this section, a brief discussion on some related works on localisation in WSN and saving energy in WSNs with clustering algorithms are introduced. The literature survey can be divided into three parts. In the first part, the localisation algorithms are discussed, in the second the clustering algorithms are discussed, and the last part focuses on MAC protocols.

Peng and Sichitiu [18] proposed ‘a new localisation and orientation scheme that considers beacon information multiple hops away, which achieves very good accuracy and precision despite inaccurate angle measurements and a small number of beacons’.

Dragos¸ Niculescu, and Badri Nath (2003) [16] proposed a method for all nodes ‘to determine their orientation and position in an ad-hoc network where only a fraction of the nodes have positioning capabilities, under the assumption that each node has the AOA (angle of arrival) capability’.

In the same year, Tian, Huang, Blum, Stankovic, and Tarek (2003) [17] proposed approximate PIT (APIT) localisation using neighbour information. Bogdanov, Maneva, and Riesenfeldi (2004) [15] improved the positions of base stations in a data-collecting sensor network. Their algorithm ‘tested to find positions which significantly improve the data rate and power efficiency of the network’.

Xiang (2004) [5] studied two issues related to sensor and object localisation in wireless sensor networks. He first examined the sensor localisation algorithms that are used to determine sensor positions in ad-hoc sensor networks. He explored the characteristics of dimensionality reduction techniques and proposed three sensor localisation algorithms based on multidimensional scaling techniques. These include a centralised sensor localisation algorithm, a distributed sensor localisation algorithm, and a robust sensor location algorithm based on multidimensional scaling.

Holly (2005) [6] proposed a dynamic version of the SpaseLoc (sub-problem algorithm for sensor localisation), which was developed for estimating moving sensor locations in a real-time environment. The method uses dynamic distance measurement updates among sensors and utilises SpaseLoc for static sensor localisation.

Yuntao (2006) [7] introduced a stochastic strategy for estimating unknown node positions in a wireless sensor network ‘based exclusively on connectivity-induced constraints, which makes this method more accurate and adaptive’.

Youssef and El-Sheimy (2009) [19] presented a process of building local WSN coordinates by considering the imperfection implicit within the estimated location angles, minimising the number of local coordinates formed within the WSN, and introduced the geometrical dilution of precision (GDOP) as a factor to distinguish between ‘the precise ranging and the precise location’.

Ziguo (2010) [8] offered novel solutions to bridge the gap between low cost and high accuracy for range-free localisation. He explored uncontrolled event-driven localisation that ‘advances the state of the art an important step towards a usable system’.

Paul and Matin (2010) [20] proposed a new optimal location for base stations using a geometrical approach for ‘maximum life of the sensor network’.

Adnan Mohammed Abu-Mahfouz (2011) [1] proposed Alwadha [1] as ‘an efficient localisation algorithm for wireless ad hoc sensor networks with high accuracy’, observing that it ‘does not depend on using a high number of references or all of them to develop the accuracy of estimation’. In fact, it utilises a smart reference selection method, that is, Alwadha selects the minimum (or nearly) possible number of references that could contribute most to high accuracy.

Finally, Carlos, Ricardo, Enrique Rodr, and Michael (2011) [9] presented a method for reducing signalling overhead due to a distributed localisation procedure.

The second part of the related work is the clustering algorithms with localisation problem. Sadegh and Mohammad (2008) [10] proposed a clustering algorithm called clustering for localisation (CFL). CFL focuses on the principles of designing a clustering algorithm in addition to providing an environment for designing a localisation algorithm based on clustering.

Yaghmaee (2010) [12] presented an efficient energy prediction based on a tracking algorithm that selects tracker sensor nodes based on both energy and distance parameters and performs localisation of the target using a trilateration algorithm. The proposed algorithm decreases ‘network energy consumption and increases the network life time’. D. Charanya and G.v.uma (2012) [14] suggested an efficient energy prediction-based clustering algorithm based on the LEACH\_R algorithm, which ‘gives better results in both performance and energy consumption’.

Hassan, Razan, Eman, and Elleithy (2013) [11] presented two object tracking localisation techniques for WSNs based on cluster algorithms that have been combined together to perform many functions. This reduces energy consumption and required communication bandwidth. Furthermore, the algorithm is highly scalable and prolongs the lifetime of the network.

The third part of the related work is focused on MAC layer protocols. Rahman, Ahmad, and Bazaz, (2012) [31] presented a TDMA-based MAC (TDMAC) protocol specially designed for applications that require periodic sensing of the sensor field. TDMAC organises nodes into clusters. Nodes send their data to their cluster head (CH), and the CHs forward it to the base station.

Smriti Joshi and Anant kr. Jayswal (2013) [28] studied an efficient energy protocol, EDMac, which features lower energy consumption than the TMac protocol. The EDMac protocol performs better than the TMac protocol as it has an additional extended activation time function, which is extracted and deduced from the original activation time function in TMac. Thus, the EDMac protocol provides perfect templates to design new high performance, contention-based WSN MAC layer protocols.

Kr. Tyagi, Maheshwari, and Upadhyay (2013) [29] proposed a new MAC protocol that uses inherent features of the TMAC protocol but optimises it with a simple reduction function. This provision cuts out the extra listening period, which has a high probability of getting waste during its period of activation.

Enam, Qureshi, and Misbahuddin (2014) [2] developed a distributed uniform clustering algorithm (DUCA) for cluster-based WSNs. In DUCA, the cluster structure technique is based on a virtual-grid system, resulting in an even distribution of clusters, homogenised cluster sizes, and reduced WSN energy consumption.

Maryam El azhari (2014) [27] stated that wireless body area networks (WBAN) provide ‘promising applications in medical monitoring systems to measure specified physiological data’ and also provide ‘location-based information’. In her paper, she also presented an overview of the existing MAC protocols, namely IEEE 802.15.4, IEEE 802.15.6, and TMAC, to highlight the prerequisites of WBAN. She also studied the performance of IEEE 802.15.4 MAC, IEEE 802.15.6 MAC, and TMAC under mobility constraint in terms of energy throughput and latency using OMNET++ with Castalia as a simulating tool. Her analysis showed that IEEE 802.15.4 outperforms IEEE 802.15.6 and TMAC with GTS ON and temporal variation.

Most algorithms use power minimisation between nodes, and in this thesis, power consumption minimisation has also been considered.

**3 Alwadha Position Clustering Algorithm**

In this section, the Alwadha position clustering algorithm is described. The algorithm is designed to estimate the accurate positions of network nodes while reducing the amount of energy consumed by the WSN. In order to achieve accurate position and minimum energy consumption simultaneously, the proposed algorithm first adopts the accurate position estimation method used by the Alwadha algorithm [1] and then integrates a clustering algorithm to produce a new method for sending node locations to the sink node.

The Alwadha algorithm consists of three stages. In the first stage, each node searches for its accurate location and determines whether to accept that location. In the second stage, if the position is accepted by the node, its location is sent to the sink following the clustering technique. If the position is not accepted, the search process is repeated to search for the accurate position. In the final stage, the nodes are categorised as being in either active or sleep mode in each round of the algorithm in order to save energy consumed by the nodes for sending, receiving, or processing.

**3.1 Position Estimation**

There are three types of nodes. The first have known coordinates and it only be a received node for the request location. The second type is the unknown node, which has unknown coordinates and can only be a requested node for the location. The final type, the known node, has coordinates that are calculated. This type of node changes from an unknown node to a known node and becomes a reference node. It becomes a received node for the request location when an unknown node or a request node develops its location when it is a known node.

The first stage of the algorithm is position estimation. Unknown nodes broadcast a message called a location request to all their first-level neighbours. First-level neighbours are those that are reachable by the current node in one hop. This message contains the value of the required accuracy level (Lacc), which is set to 0 for unknown nodes. For known nodes, the accuracy level is equal to the lower probability of accuracy in the subset Si that they used to estimate their locations, as will be explained later.

References which receive the location requests reply by location response, which contains the probability of accuracy and the location. However, the references will not replay the known request node until they check whether their probability of accuracy is higher than the demand accuracy level of the request node and higher than the probability of response (0.57). The probability of accuracy of the references is equal to 1, and for the known node it is equal to:



References Ri (which may be reference or known nodes) send location responses to the asked node, and the location response contains reference ID, location, and probability of accuracy. Then, the smart reference selection method is applied to support the node in choosing the most accurate subset of references (Si) and estimate its probability of accuracy. The smart reference selection method begins when the node receives the response message from references Ri. It groups them in decreasing order depending on the probability of accuracy, which is used to choose a further subset (Si). Next, the node chooses the first three references in Ri to Si and then measures the probability of the accuracy of Si using the following equation:



where Pacc of j is the probability of accuracy of reference Ri.

If the calculated Pacc is lower than a definite value (Pmin = 0.5), the following reference should be added to Si and then Pacc is recalculated. The node reloads with this operation until the subset Si accepts a particular value of probability of accuracy (Pacc >= Pmin), or it ends and postpones the estimation to the following repetition. Next, the RSS mechanism is used to measure the distance from the node to each reference. Then, the node uses the minimum mean square estimate (MMSE) to determine the initial position of the unknown node.

Following this, the node develops the accuracy of position estimation by examining the possibility of development based on the estimated distance error Ei,j, which is the distinction between the calculated distance (between the node’s initial position Zi and the reference position Zj) and the measured distance (Di,j) as determined by the following equation [1]:

Ei,j = | || Zi - Zj || - Di,j| ,

where j € Si. If one or more of the references has an estimated distance error greater than the specific maximum value (Emax = 1.0), such as (Ei,j > Emax), that must develop the position. Otherwise, the initial position continues as an accurate position and the operation goes to the following stage. To develop the position estimation, the node removes references that have (Ei,j > Emax), and a new subset of references will then be selected to estimate a refined position, as described in the smart reference selection method. Finally, the node calculates the position’s degree of accuracy (Dacc) as follows:

Dacc = Σ ∣ ∥ Zi − Z j ∥− Di,j ∣ ,

where j ∈ S.

Each iteration k calculates the degree of accuracy, and if the current degree is better than the last one (Dacc(k) < Dacc(k−1)), the node accepts the estimated position; otherwise, the estimated position is rejected. This operation is performed to check whether the accepted position is more accurate than the previous position.

Finally, the node checks whether the degree of accuracy is less than the accuracy target degree Tacc (Dacc < Tacc). It considers this position as the accepted position and finishes the location request message operation. The accuracy target Tacc = 0.4.

Now each node knows its accurate location and does not broadcast it to the whole network but behaves in a clustering way to save more energy in the WSN.

**3.2 Clustering Mechanism**

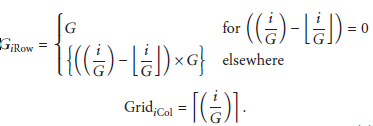
In the first stage, the node finds its accurate position. In this stage, the node wants to save energy for the WSN when the node knows its position, applying a clustering item to the model. In a large scale WSN, the nodes usually collaborate with each other to collect and forward data in a hierarchal manner. For this purpose, cluster-based protocols are commonly employed to reduce the energy consumption of WSNs.

In this stage fetch, the CHs will done. And, to fetch an equal distribution of CHs, the network must be split into virtual grids.

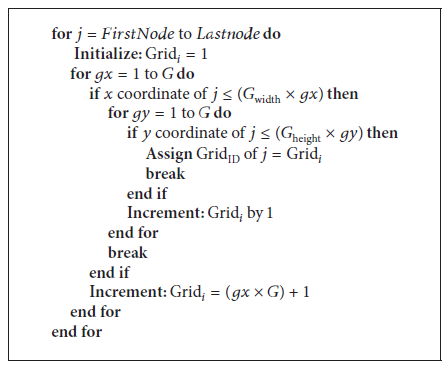
First, the sink (base station) determines the desired number of CHs (𝐶) in the space; to divide the CHs evenly, the network region is split into 𝐺𝑇 number of grids. The network is diffused on a square region with the dimensions of 𝑋 by 𝑌 units. The sink computes and gives grid IDs to all sensor nodes in the network, based on the following steps:

(1) The sink splits the network region into a similar sized ‘number of virtual grids’. Each grid has an ID according to its row number and column number, computed as follows:

Let 𝐺 = grids in one row = grids in one column. The height and width of each grid are 𝐺height = 𝑌/√𝐺𝑇 and 𝐺width = 𝑋/√ 𝐺𝑇 [2]. Based on the row number and column number of each grid, their grid IDs are computed using the following, where 𝑖 is the grid ID of the 𝑖th grid:



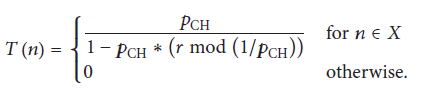
(2) Each node transmits its location information (𝑥, y) coordinates, which are computed in the first stage, to the sink node. The sink node uses this mechanism to compute the grid ID for each node, as shown in Figure 3.1 [2].



*Figure 3.1*. Grid-IDs to each node algorithm.

(3) Next, the sink broadcasts the grid IDs of all nodes.

In the end, every node knows its location and to which grid it belongs, and it knows it must fetch a fair selection of CHs to balance the energy consumption between all nodes in the network. Therefore, the CHs are picked based on the number of times they have been selected in the former rounds. This is done in each node, which generates a random number between 0 and 1, and if the number is less than a threshold value (𝑛), the node fetch selected for that round. The value of (𝑛) is computed as [2]:



𝑝CH is the required percentage of cluster heads in the network; that is,

𝑝CH = 𝐶/𝑛, where 𝑛 is the total number of nodes in the network and 𝑋 is the set of nodes that have not been CHs in the last 1/𝑝CH rounds. 𝑟 is the round number for which the CH is being chosen [2]. Now, the CHs are distributed, but not evenly because fetching an even distribution of CHs in the network requires the following steps:

1- The CHs are chosen that knew their grid region and broadcast their residual energy levels and their grid IDs to the nodes in their grid region.

2- After broadcasting in step 1, if there is more than one CH in a grid, then the node which has the highest energy level fetches itself automatically. The remaining CHs in the same grid will return to their normal node mode for this round.

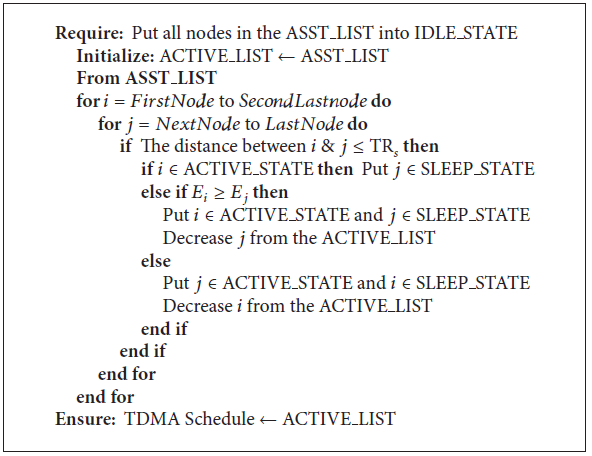
3- Finally, the selected CHs announce their elections because the broadcast range grows to cover the nodes in the adjacent grids. This operation occurs because there are grids with no CHs in them. Therefore, they associate themselves with the closest CH in an adjacent grid [2].

**3.3 Node Association**

Finally, the remaining nodes define their closest CH by the received signal strengths of multiple CHs. This stage uses threshold sensing range (TRs) parameters to decrease the sizes of the clusters, depending on the sensing range of the nodes, as follows:

1- Nodes broadcast their association requests to the CH, containing their residual energy levels and their locations to the respective CHs. The CH is the shortest distance to the nodes.

2- CHs compute the distances between each of their associated nodes. Then, they choose the active nodes for this round according to the algorithm shown in Figure 3.2 [2].



*Figure 3.2*. Select associate nodes algorithm.

The asst list is the list of nodes that are associated to this CH, and the active list is the list of nodes that are active for this round only.

3- The CHs set a node to either a sleep or active state depending on the TRs value between the nodes and the energy level 𝐸 of the node. Finally, the CHs support the slots in the TDMA schedules only for the active nodes while the excessive nodes remain in sleep for this round (Figure 3.2) [2].

This chapter presents the proposed algorithm, the Alwadha position clustering algorithm for WSNs, and information on how it is designed.

**4 The Effects of MAC Protocols on Alwadha Algorithm Performance**

This chapter presents the first attempt to enhance the level of node energy consumption by considering the effect of using different MAC protocols on the energy consumption level of the Alwadha algorithm. Sensors are battery-operated nodes that communicate with each other wirelessly in a WSN. The major issue related to communication is energy consumption because there are areas that are difficult to reach, making it difficult to recharge the batteries. Using a suitable MAC protocol would help to decrease the power consumption level and consequently increase the network’s lifetime.

Saving energy is a crucial factor for almost all protocols used in WSNs, particularly MAC protocols. These protocols deal directly with the state of energy consumption in either a positive or negative way. The MAC layer is determined to be an important source of energy waste based on the following symptoms:

1. *Overhearing:* where each node receives every packet that is sent by other nodes. Receiving unnecessary radio signals increases the energy consumption level.
2. *Collisions:* where there are many nodes sharing the radio channel at the same time. When nodes send their packets, collisions occur. This increases the number of latency packets and also energy consumption because a retransmission mechanism is needed.
3. *Control packets:* the MAC protocol uses control packets such as request to send (RTS), clear to send (CTS) an ‘acknowledgement’ (ACK), and headers. These packets do not contain any application data, so they are considered as pure overhead.
4. *Idle listening*: this situation occurs when nodes are not active and are trying to receive when nobody is sending. In this situation, the amount of energy wasted is similar to the energy wasted by a common reception.
5. *Over-emitting*: when nodes send data to receiver nodes when those nodes are not ready to receive, hence wasting energy.
6. *Traffic fluctuation*: variations of the traffic load can affect the waste of a node’s energy reserves. Thus, adaptive traffic should be in the protocol.
7. *Flooding*: when sensor nodes send their data packet to the root node simultaneously. In this case, the flooding has occurred and results in collision and loss of data packets, and retransmission requests of the lost data packets from the nodes are then needed.

In order to decrease energy waste, several protocols has been proposed, which may be categorised into two main classes:

1. Scheduled-based Protocols: These protocols, which are recognised as deterministic, work to avoid collisions by associating a slot time for a sensor node in a specific cluster, and to relieve the belongings of the overheating problem, as in this situation each node determines the identical slot time to transmit its data packet.
2. Contention-based Protocols: These protocols, which are recognised as CSMA-based (carrier sense multiple access), are commonly used in multi-hop wireless networking due to their simplicity and sufficiency, that is, their ability to be executed in a decentralised environment such as a WSN. To reduce collisions and to decrease other common sources of energy waste, the wake-up–sleep mechanisms and the control message RTS and CTS an ACK used in the 802.11x standard are used to design energy-efficient MAC protocols for WSNs, such as S-MAC, TMAC, TunableMAC, and B-MAC.

**4.1 TMAC Protocol**

Timeout medium access control (TMAC) allows WSNs to work on their radios at synchronised times, and to work them off after a definite period during which no communication takes place. The TMAC protocol wakes up to transport with its neighbour nodes and then returns to sleep until the following frame. At the same time, there is a queue of fresh messages. Sensor nodes communicate with each other using an ACK, RTS, CTS, and data planner, which supply collision avoidance and reliable transmission. A node will hold listening and transporting potential as long as it is in an active period. The active period ends when no activation event has happened for a particular active time (TA).

In TMAC, all messages are transported in a burst of changing lengths, and there is a gap between the bursts called sleep–sleep time, which decreases idle listening. In Figure 4.1, the TA finishes when there is no active event for a time period TA, and the node shifts to sleep mode. At the time of top load, the node communicates continuously without sleeping.



*Figure 4.1*. Nodes with active time TA

**4.2 TunableMAC**

TunableMAC is designed with many parameters. With suitable tuning, it can simulate many other communal MAC protocols. It is designed for broadcast transportation, and it does not support acknowledgments.

It is a contention-based protocol and uses the CDMA technique to relieve packet collisions. It also upholds duty-cycling, which supports the node to turn off the radio and sleep during ideal periods to conserve energy. Several key parameters determine its operation:

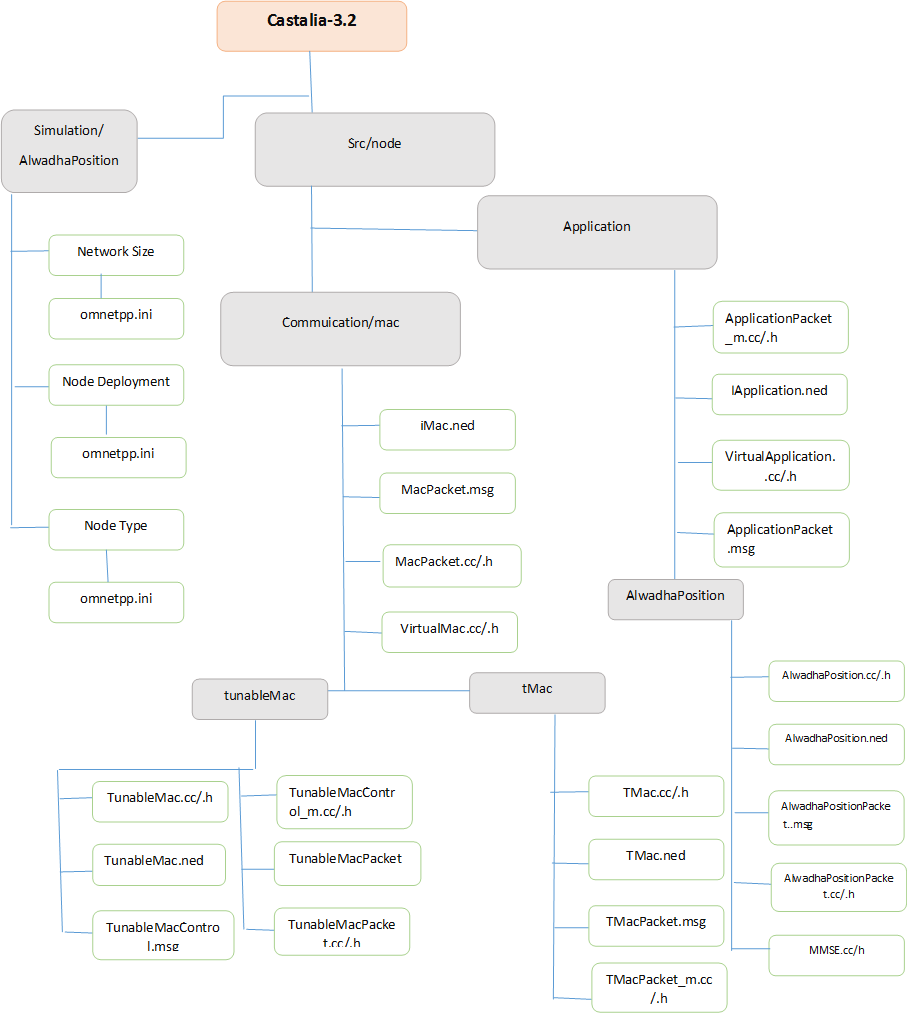
* Duty Cycle: This parameter is the time that the radio pauses and the node listens to the channel. It directly impacts node energy consumption.
* Listen Interval: This parameter is how long, in milliseconds, the node listens to the channel through a single period. The listen interval, combined with the duty cycle, defines the length of a single period as the listen interval/duty cycle.
* Beacon Interval: Nodes send out a set of beacons to their sleeping neighbours to wake them. This parameter defines how long nodes must transmit their beacons.
* CSMA persistence: A value of 0 indicates a non-constant CSMA, so nodes back off when the channel is busy.

**4.3 Implementation**

In this thesis, TMAC and TunableMAC protocols will be applied under the Alwadha model in order to reduce the energy consumption of each sensor and prolong network lifetime.

**4.3.1 The Structure of the Model in Castalia-3.2**

In this section, the model resulted from combining the Alwadha position model and the two MAC protocols, the TMAC model and TunableMAC: The Alwadha position model is derived from the Alwadha localisation algorithm while the TMAC model and TunableMAC is derived from the MAC layer. The TMAC model includes TMac.cc/.h files, TMacPacket\_m.cc/.h files, TMac.ned, and TMacPaket.msg. This model implements the Alwadha algorithm in the application layer and applies the TMAC protocol in the MAC layer. The TunableMAC model includes TunableMac.cc/.h files, TunableMacPacket\_m.cc/.h files, TunableMac.ned, and TunableMac.msg. This model implements the Alwadha algorithm in the application layer and applies the TunableMAC protocol in the MAC layer. Figure 4.2 shows the structure of the model in Castalia-3.2.



*Figure 4.2*. Structure of Castalia.

**4.3.2 Simulation and Evaluation**

This section evaluates the Alwadha algorithm, looking at the effects of node deployment, node density, and network size on energy consumed. It will examine the impact of TMAC and TunableMAC on the application of the Alwadha algorithm.

**4.3.2.1 Environment**

The Alwadha localisation algorithm was evaluated using the WSN simulator (Castalia-3.2). The configuration file starts with a general section. The first line in the general section includes the command Castalia.ini, which assigns some parameters affecting general OMNeT execution as well as parameters that map the different random number generators (RNGs) to modules. The simulation time is set to 10 seconds. The initial network parameters for SN are defined, such as field size, the number of nodes, and the deployment type. After these top-level parameters are defined, configuration-related module parameter settings are defined.

**4.3.2.2 Results**

Several experiments were performed to evaluate the energy consumed using the Alwadha localisation algorithm and applying different MAC models. Three measuring criteria parameters were used: node deployment, node density, and network size. In each experiment, these parameters were evaluated and compared based on the energy consumed.

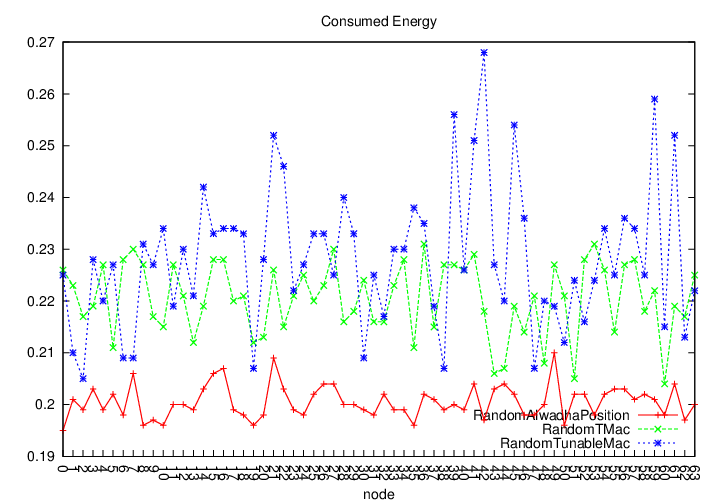
**4.3.2.3 Node Deployment**

Table 4.1 shows the setup environment setting and node deployment factors for every experiment. In the first experiment, the sensor nodes were deployed randomly, and in the second, grid deployment was used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factor | Fig. | Deployment | Area | #Beacon | #Unknown |
| Node Deployment | 4.3 | Random | 200x200 | 12 | 52 |
| 4.4 | Grid | 200x200 | 12 | 52 |

*Table 4.1*. Node deployment environment.

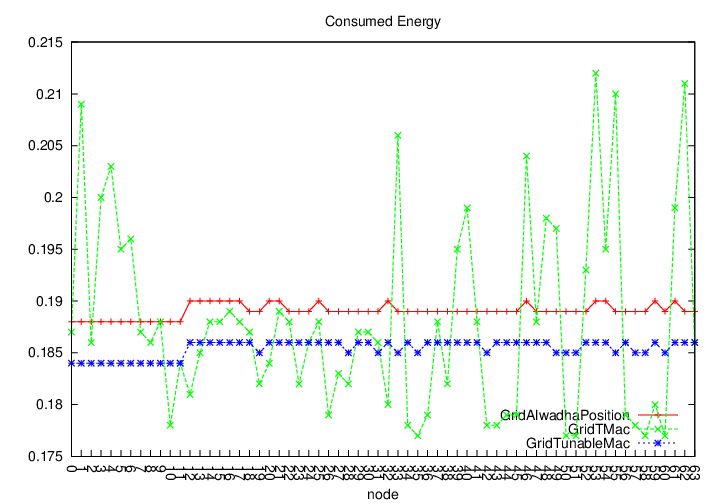
In this experiment, nodes were deployed randomly. Figure 4.3 presents the energy consumption results of applying the Alwadha position algorithm, the Alwadha application with the TMAC model, and the Alwadha application with the TunableMAC model.



*Figure 4.3*. Random deployment.

The results show that when the Alwadha algorithm was applied with no MAC protocol and in a random deployment environment, the total energy consumption was lower than when using the algorithm with both TunableMAC and TMAC protocols. The total energy consumption is a summation of the energy consumed by all nodes included in the network.

In the second experiment, nodes were deployed in a grid network. The changes in total energy consumption versus the number of nodes are presented in Figure 4.4.



*Figure 4.4*. Grid deployment.

The use of TunableMAC reduced energy consumption more than the use of the TMAC model. For the TunableMAC, the results when using grid deployment were better than when using random deployment. This is because in grid deployment most nodes have the same number of neighbours, whereas in random deployment the nodes could have a different number of neighbours. A fixed number of neighbours leads to lower and stable energy consumption. Another factor that lowered energy consumption was the TunableMAC’s fixed duty cycle design. The energy consumption using TMAC was similar to the case of random deployment because of its dynamic duty cycle design.

The results of these two experiments showing the average energy consumption for each experiment are summarised in Table 4.2

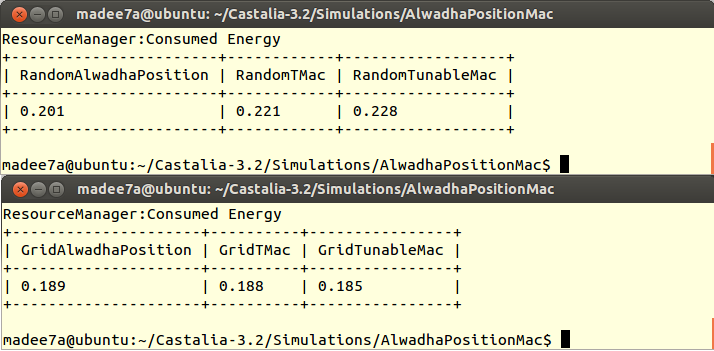


Table 4.2: Average Energy Consumption of Node Deployment

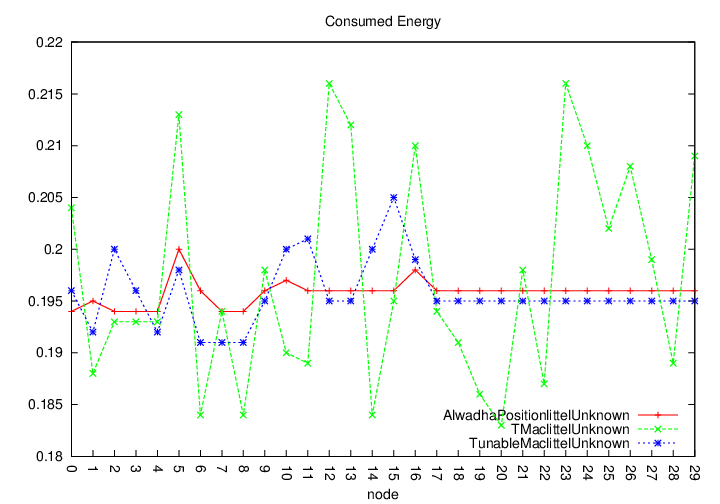
**4.3.2.4 Node Density**

The WSN topology consists of two types of nodes: beacons and unknown nodes. This part of the results checks the effect of changing the number of these nodes on the performance of the localisation algorithms. First, a fixed number of beacons were used while changing the number of unknown nodes, then the number of unknown nodes was fixed while the number of beacons was changed. In all the experiments reported in this section, the nodes were deployment randomly in a 200 m × 200 m field, as shown in Table 4.3.

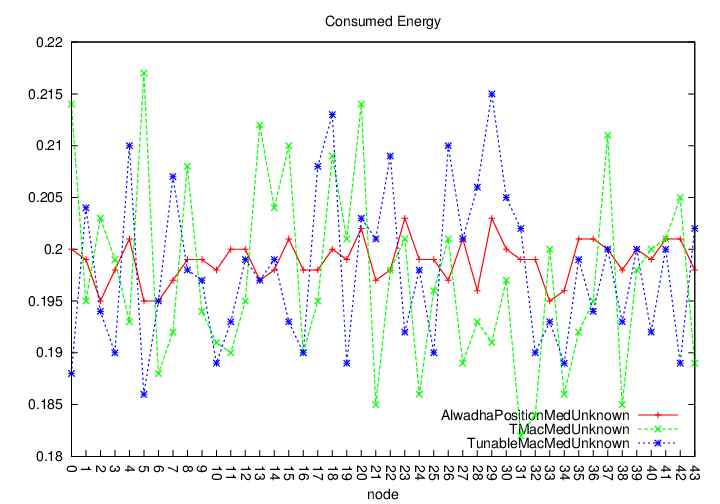
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factor | Fig | Deployment | Area | #Beacon | #Unknown |
| Node  density | 4.5 | Random | 200x200 | 9 | 21 |
| 4.6 | Random | 200x200 | 9 | 35 |
| 4.7 | Random | 200x200 | 9 | 51 |
| 4.8 | Random | 200x200 | 9 | 21 |
| 4.9 | Random | 200x200 | 23 | 21 |
| 4.10 | Random | 200x200 | 39 | 21 |

Table 4.3: Node Density Environment

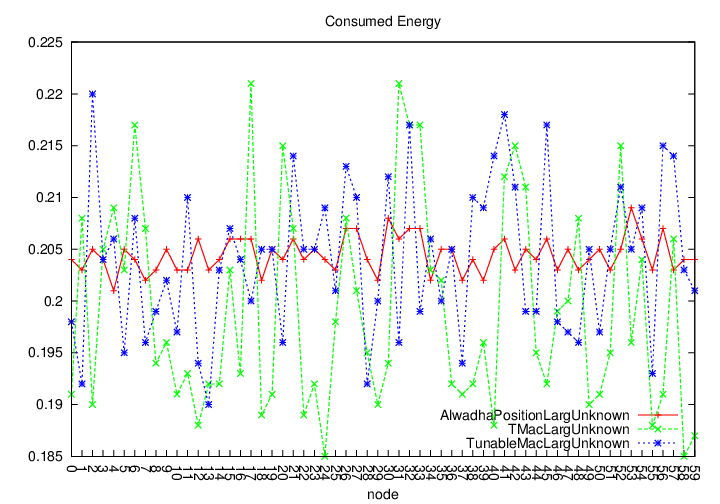
In the first three experiments, the nodes were deployed randomly. Figures 4.5, 4.6, and 4.7 present the energy consumption of the original localisation algorithm (Alwadha), the Alwadha application under the TMAC model, and the Alwadha application under the TunableMAC model (Figure 4.5 a small number of unknown nodes, Figure 4.6 a medium number of unknown nodes, and Figure 4.7 a large number of unknown nodes). To examine the density of the unknown nodes, 9 beacons were used, and the unknown nodes’ density varied as 21, 35, and 51 unknown nodes.



*Figure 4.5*. Little unknown density.



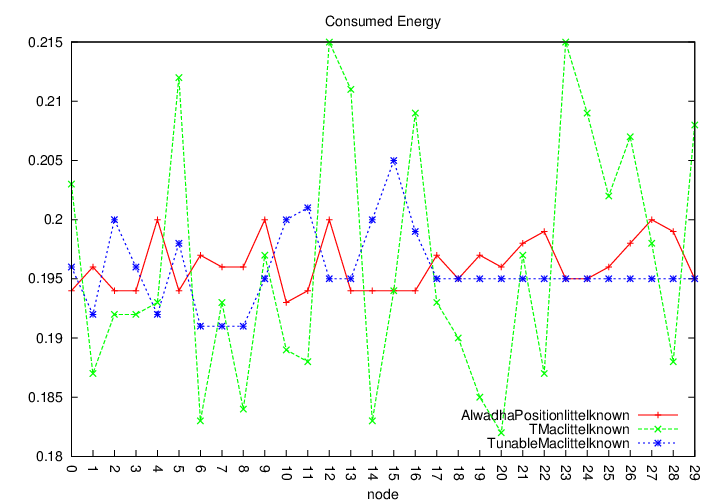
*Figure 4.6*. Medium unknown density.



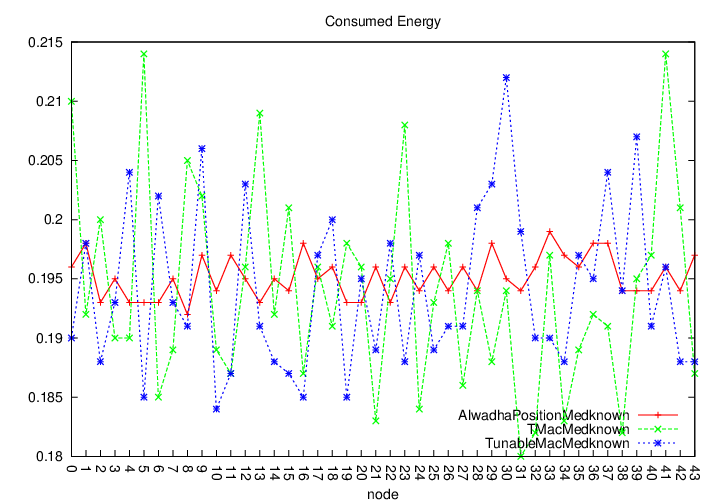
*Figure 4.7*. Large unknown density.

From Figures 4.5, 4.6, and 4.7, it can be seen that when using different unknown nodes’ density, the smart references selection method used by the Alwadha algorithm enabled the nodes to estimate their location with the best accuracy, but at the same time increased energy consumption compared to the other models. The energy consumption with TunableMAC increased for frequent listening when no events occurred due to increases in the density of the unknown nodes. Implementation of the TMAC protocol with unknown node density increased slowly to keep energy consumption low using future request to send (FRTS) to avoid the early sleeping problem.

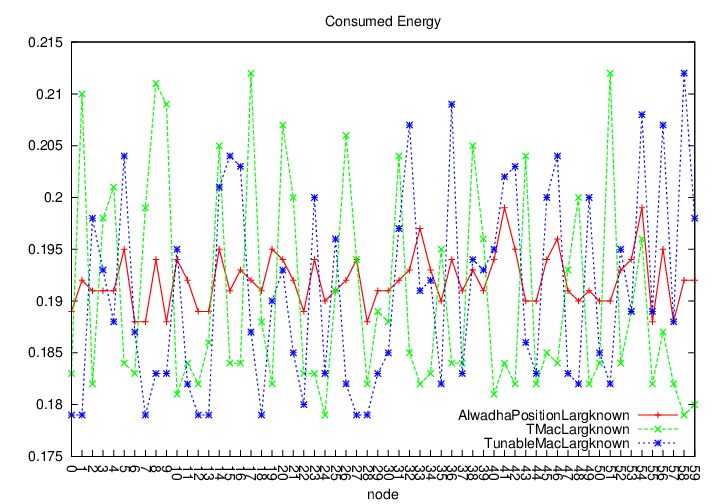
In the last three experiments, the nodes were deployed randomly. Figures 4.8, 4.9, and 4.10 present the energy consumption of the original localisation algorithm (Alwadha position), the Alwadha application under the TMAC model, and the Alwadha application under the TunableMAC model (Figure 4.8 a small number of known nodes, Figure 4.9 a medium number of known nodes, and Figure 4.10 a large number of known nodes). To examine the known nodes’ density, 21 unknown nodes were used, and density varied as 9, 23, and 39 known nodes.



*Figure 4.8*. Little known density.



*Figure 4.9*. Medium known density.



*Figure 4.10*. Large known density.

From Figures 4.8, 4.9, and 4.10, when using different known densities, the Alwadha algorithm achieved the best accuracy. An interesting observation about Alwadha is that the average number of references used is reduced by increasing the beacon density. This is due to the smart reference selection mechanism, which results in more references with a high probability of accuracy, enabling the nodes to select a smaller subset of references following the condition Pacc >= Pmin. The energy consumption of TunableMAC is low when the network is idle due to increases in the density of known nodes, which means few nodes need to know its location. The implementation of the TMAC protocol on known nodes’ density increases slowly to keep energy consumption low, and it uses FRTS to avoid an early sleeping problem when the network is idle.

The results of these six experiments, which show the average energy consumption for each experiment using different unknown nodes and beacon density, are summarised in Table 4.4.

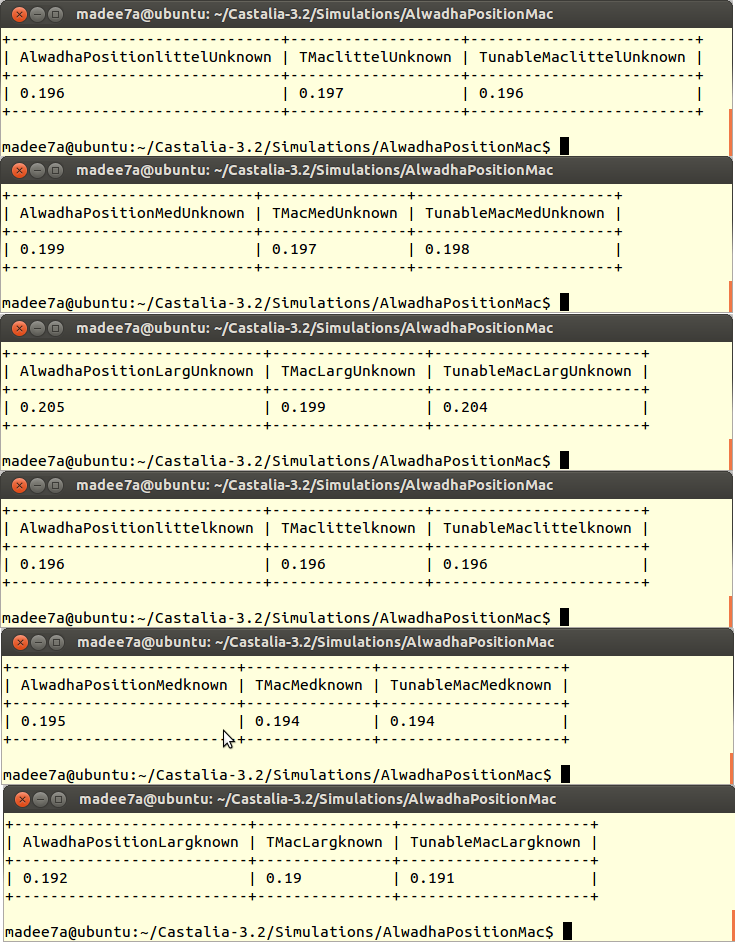


Table 4.4: Average of Energy Consumption of Node Density

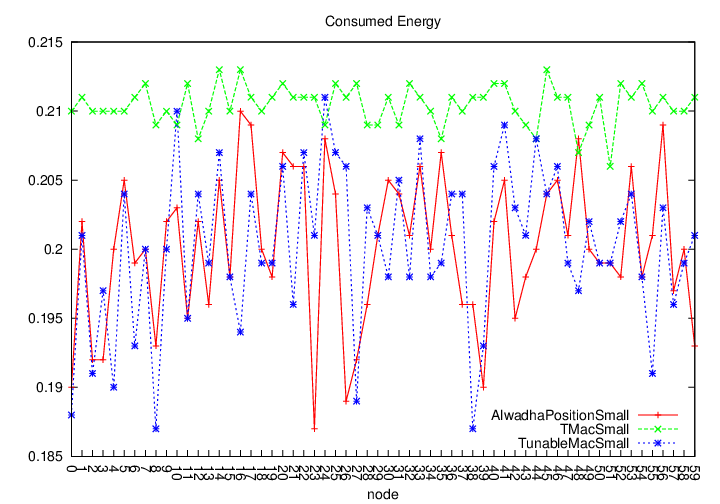
**4.3.2.5 Network Size**

To determine the effect of changing the network size on the performance of the Alwadha localisation algorithm with MAC models, three experiments were performed using different network sizes (small, normal, and large), as shown in Table 4.5.

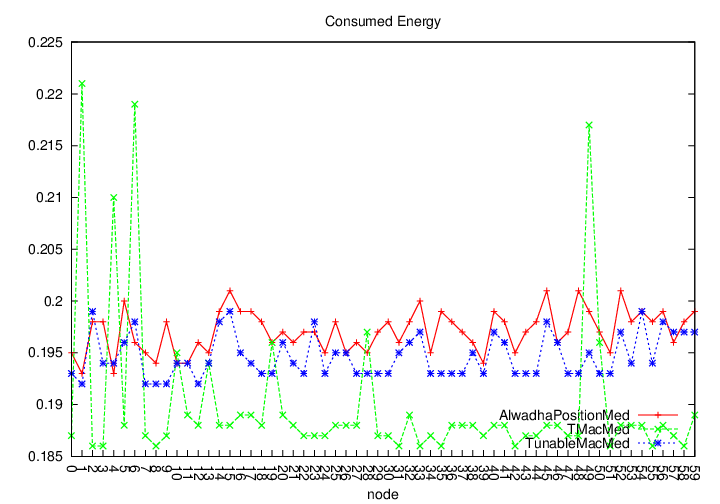
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factor | Fig | Deployment | Area | #Beacon | #Unknown | Notes |
| Network  Size | 4.11 | Random | 100x100 | 9 | 51 | Small |
| 4.12 | Random | 200x200 | 9 | 51 | Normal |
| 4.13 | Random | 600x600 | 9 | 51 | Large |

Table 4.5: Network Size Environment

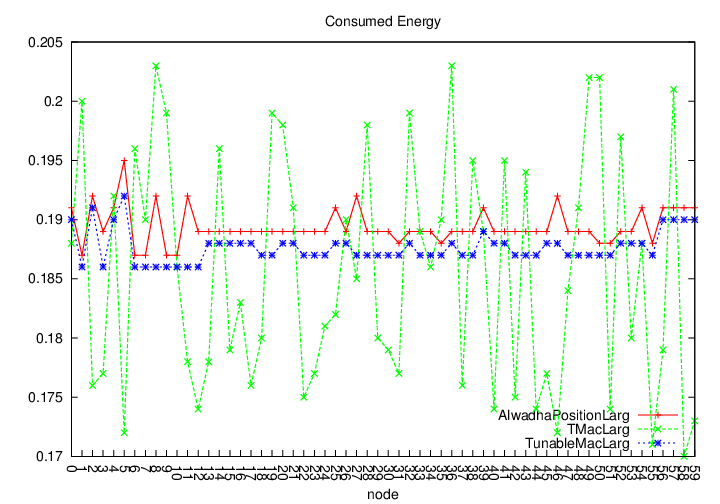
In these three experiments, the nodes were deployed randomly in differently sized networks. Figures 4.11, 4.12, and 4.13 present the energy consumption of the localisation algorithm (Alwadha position), the Alwadha application under the TMAC model, and the Alwadha application under the TunableMAC model (Figure 4.11 a small network, Fig 4.12 a medium network, and Figure 4.13 a large network).



*Figure 4.11*. Small network size.



*Figure 4.12*. Medium network size.



*Figure 4.13*. Large network size.

From Figures 4.11, 4.12, and 4.13, when using different network sizes, the accuracy of the Alwadha algorithm was almost the same in the three experiments. At the same time, however, energy consumption increased compared with the other models. The energy consumption using TunableMAC increased, and more collisions occurred while some packets were dropped due to the fixed duty cycle. Implementation of the TMAC protocol resulted in increased traffic loading on the network, causing more collisions, although the TMAC model overhearing avoidance control packet (RTS/CTS/DATA/ACK) was useful for collision avoidance and decreasing the amount of energy consumed.

The results of these three experiments, which show the average energy consumption for each experiment using different network sizes, are summarised in Table 4.6.

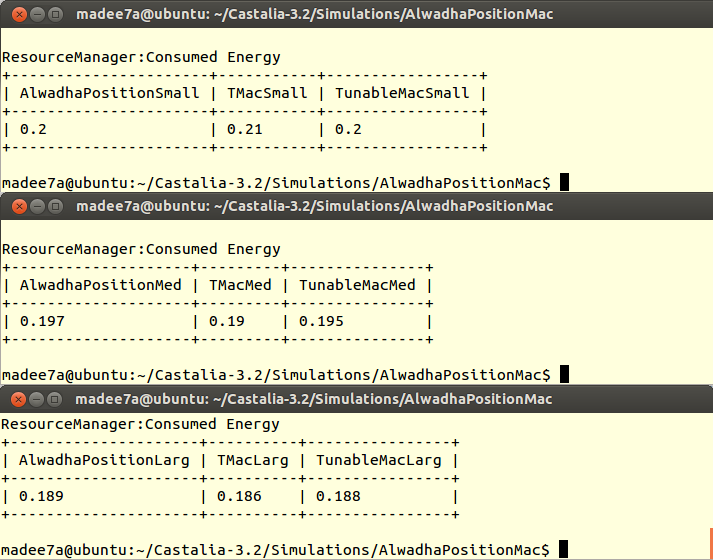
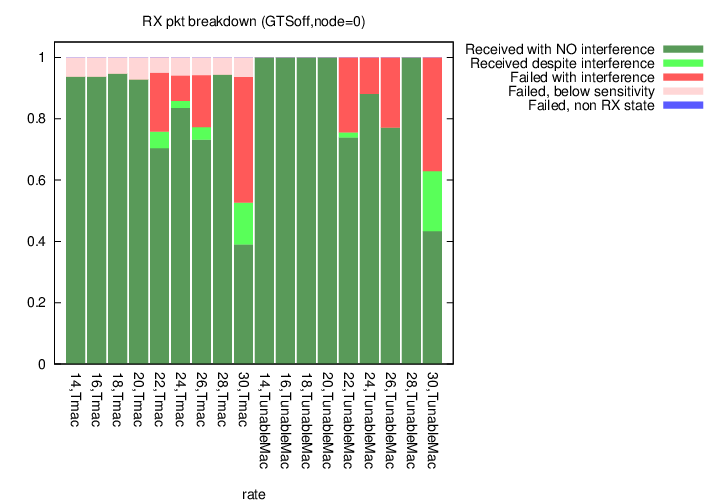


Table 4.6: Average of Energy Consumption of Network Size

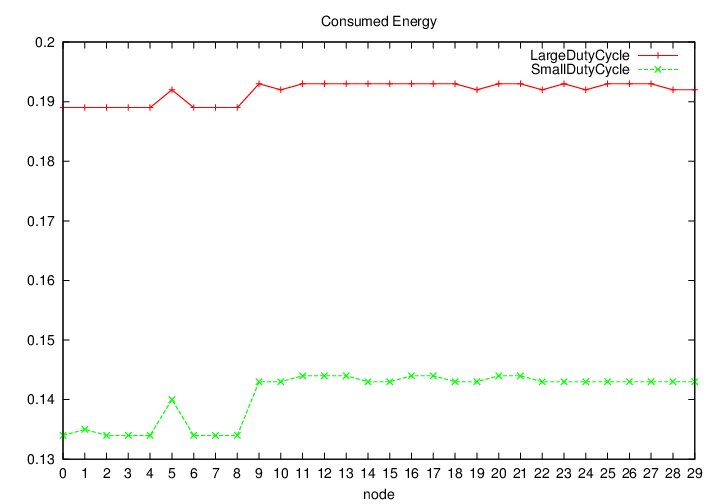
**4.3.2.5 Performance Evaluation Due to the Use of MAC Protocols**

* The performance of WSNs when using the Alwadha position algorithm under the TMAC protocol may be affected, with some messages getting lost because of the early sleeping problem, where nodes may sleep according to their activation time, as shown in Figure 6.2.



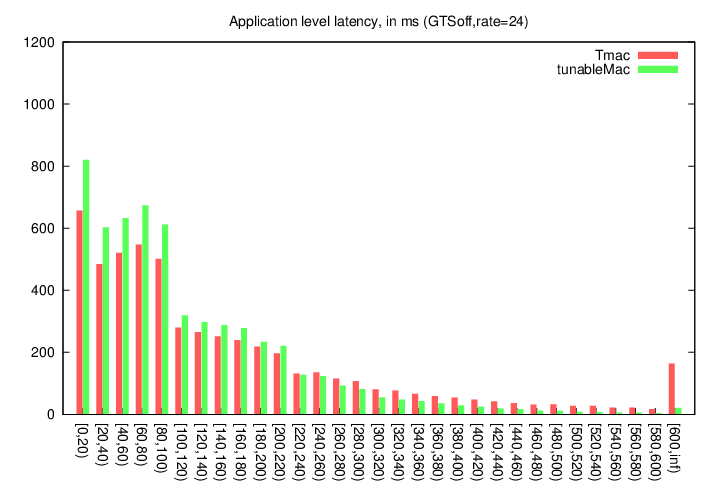
*Figure 6.2*. RX packet breakdown in the TMAC and TunableMAC models.

* The network load dynamically changes, and thus the fixed duty cycle of TunableMAC cannot cause the node’s activity time to reflect the data transmission requirements of the network, resulting in an increased end-to-end delay.
* The performance of WSNs using the Alwadha position algorithm under TunableMAC with a fixed duty cycle can be improved by using no events, but packets may be dropped due to queue overflow when an event occurs. To decrease this packet drop, the fixed duty cycle should be expanded, but this results in wasted energy for frequent listening when no events occur. Thus, it is desirable to expand the duty cycle to prevent packet drop under heavy traffic and reduce the duty cycle for nodes that are unaffected by the traffic or are under light traffic to save more energy, as shown in Figure 6.3 below.



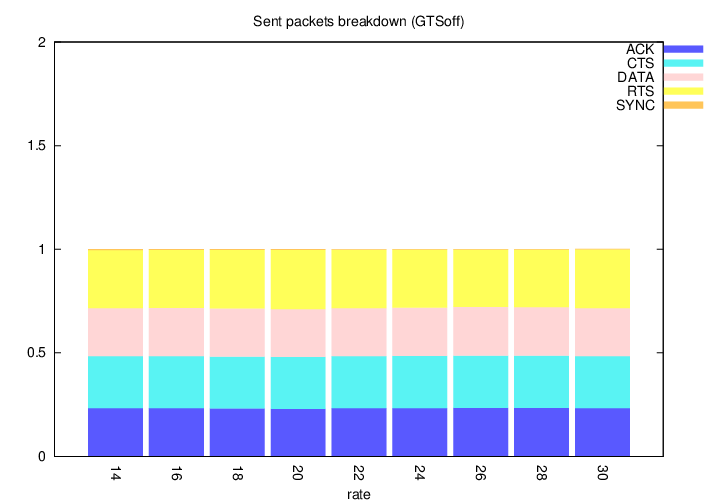
*Figure 6.3*. Different duty cycle of TunableMac model.

* TMAC can increase the duty cycle significantly when the traffic load is high. Meanwhile, the sender and receiver can achieve duty cycle synchronisation. Therefore, TMAC can have lower transmission latency than TunableMAC, as shown in Figure 6.4.



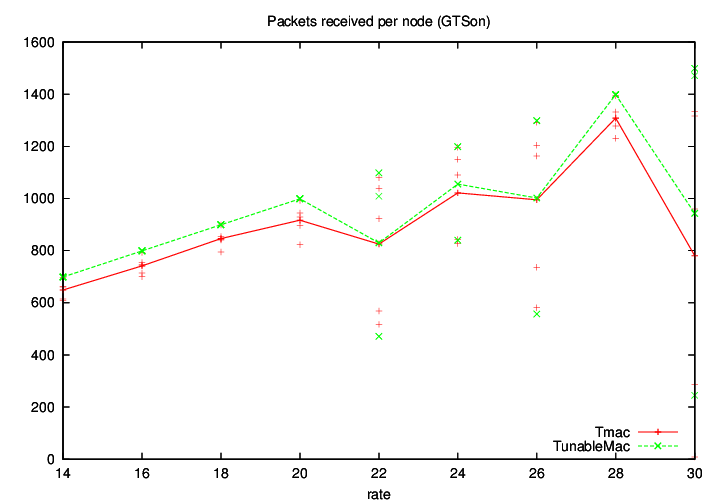
*Figure 6.4*. Application level latency of TMAC model.

* In linear topology, the proposed TMAC can have better performance than TunableMAC because TMAC can adjust a sensor’s duty cycle efficiently. Meanwhile, the control packet can make the sender and receiver have the same duty cycle, which can reduce the energy wasted in idle listening, as shown in Figure 6.5 below.



*Figure 6.5*. Control packets of the TMAC model.

* After the duty cycle is synchronised, the sender can send data to the receiver without mis-scheduling, and the data throughput will then grow rapidly. As the traffic load increases, the duty cycle of the forwarding node approaches the top duty cycle value. Then, the data throughput will be reduced as the forwarding node cannot get more time to send data. TunableMAC uses a specific duty cycle layout to send data packets whether or not the traffic load is high. In TunableMAC, the data throughput remains in a constant state, as shown in Figure 6.6.



*Figure 6.6*. Throughput level of TMac and TunableMac models.

**4.4 Conclusion**

First, I presented an overview of MAC protocols and why they are important to reducing energy consumption. Next I compared three approaches for saving energy in sensor networks: the Alwadha algorithm, the Alwadha algorithm under the TMAC model, and the Alwadha algorithm under the TunableMAC model. The Alwadha algorithm has good accuracy and uses a low number of references, a smart reference-selection method, and a termination creation approach. This reduction in the number of references greatly improves computation, communication, and energy consumption. When applying the TunableMAC model, energy consumption was enhanced, but there were some problems, such as when a traffic load was increasing and the fixed duty cycle increased energy consumption. However, it is a good model when the network is idle or when there are no events as it gives the best energy consumption value. The TMAC model saves the most energy when there is a traffic load, when the network is large, or when there is a high density of nodes because it uses numerous approaches to saving energy, such as a dynamic duty cycle, control packet RTS, CTS, ACK, and FRTS to avoid collisions and overhearing. The Alwadha algorithm under the TMAC model resulted in more energy savings in these experiments than the other models.

:

**5 Implementation**

The proposed integration technique was implemented based on its efficiency. The implementation could be performed using one of two options: real deployment of the WSN or using a WSN network simulator. In this work, we excluded the first option due to cost. Therefore, the proposed work was implemented using a WSN simulator. There were several candidates that could have been used to implement our work, however there are criteria that need to be followed in selecting the most suitable choice. According to our work, the selected simulator should provide realistic implementation of the wireless channels and the node power consumption models to ensure valid results similar to a real environment. One additional selection criterion is that the simulator had to be known to the academic community and allow for the addition of new modules such as the protocol proposed in this dissertation.

Based on the previous discussion, Castalia/OMNet++ and SN2 were determined to be two suitable candidates, but Castalia/OMNet++ was preferred for the following reasons:

1. Even though The NS-2 simulator is the most commonly used simulator in whole networking research, its learning curve is steep and much effort is required to understand how to modify the protocol.
2. Castalia/OMNet++ is a well-designed component-based simulator. It allows for easier addition or modification of protocols, and its execution is much simpler than that of NS-2.

**5.1 OMNet++ Simulation**

The OMNet++ simulation engine depends on simple modules. These modules are the main building blocks that are used in forming the complex simulation modules, with each complex module containing various simple modules. The OMNet++ model consists of a hierarchy of nested modules. Modules communicate through a message passing technique, and they are linked through gates. Messages use complex data structures and can go to their destinations directly or through intermediate modules. Modules receive messages from other modules or from within that module itself. Gates are the interface through which modules communicate. Various parameters of the module are determined either in the NED file or in the network configuration file.

**5.2 Castalia**

Castalia uses the main properties of OMNet++ to create a simulation platform particularly geared for WSNs. One of the main features of Castalia is its almost realistic wireless and radio characteristics. Castalia is built on top of the OMNet++ simulator, using its modular architecture; it retains the main architecture of OMNet++ but has some extra features. Simulators are an important tool for examining WSN applications before using them in SN.

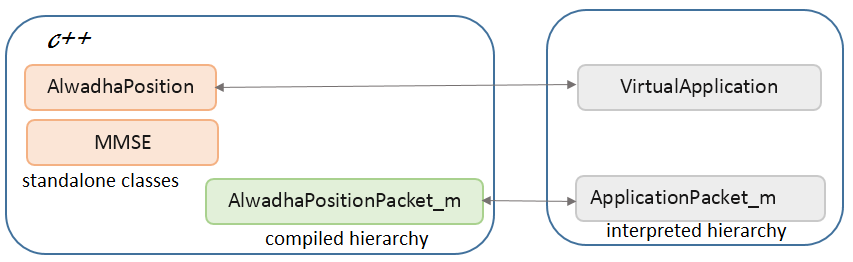
Castalia is a simulator for WSNs based on the OMNet++ simulator. It simulates real wireless environments including the channel and the node power model behaviour in an accurate way.

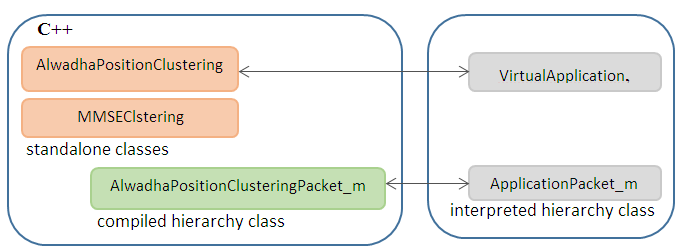
**5.3 Simulation**

To implement and evaluate the proposed localisation algorithm, new modules were added to Castalia version (3.2). The first module represents the original Alwadha position algorithm, and the second one is the new integrated Alwadha position clustering algorithm.

Figure 5.1 shows the new classes added to the basic Castalia model to represent the Alwadha position model. These classes are classified into two categories. The first includes a couple of standalone classes: MMSE and Alwadha position. These classes are used only from the C++ domain. The second category includes a compiled hierarchy class, which represents the AlwadhaPositionPacket\_m class [1].

Figure 5.2 shows the new classes that were added to the Castalia application for the Alwadha position clustering model. These classes can be divided into two types. First, there are standalone classes: MMSEClustering and AlwadhaPositionClustering classes*.* These classes are used only from the C++ domain. Second, there is a compiled hierarchy class, the AlwadhaPositionClusteringPacket\_m class. In order to access these classes from the .ned domain, they should be linked to the corresponding interpreted hierarchy classes, Application/VirtualApplication, ApplicationPacket\_m.

*Figure 5.1*. AlwadhaPosition classes added to Castalia.

*******Figure 5.2*. AlwadhaPositionClustering classes added to Castalia.

***MMSE and MMSEClustering classes***

These classes are accountable for all the mathematical matrices operations required to solve localised equations using the multilateration method [1]. Instead of using a general matrices multiplication and matrix inverse, optimised methods dedicated mainly to MMSE were implemented. These optimised methods required less computation and a shorter execution time.

***AlwadhaPosition and AlwadhaPositionClustering classes***

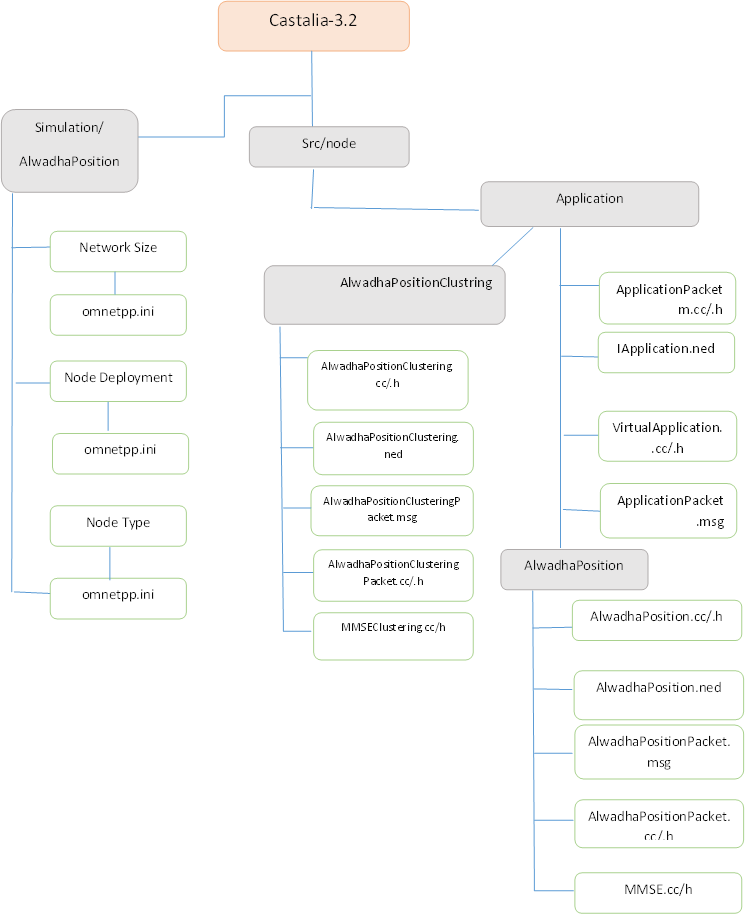
These classes are derived from the Virtual-Application class and include the implementation of an Alwadha localisation algorithm. The AlwadhaPosition class has more functionality, such as a smart reference-selection method, specifying the number of references used, and applying a termination criterion, as explained earlier. The AlwadhaPositionClustering class includes the implementation of the Alwadha position clustering algorithm, as explained in Chapter 3.

***AlwadhaPositionPacket\_m and AlwadhaPositionClusteringPacket\_m classes***

These classes were generated by alwadhapositionPacket.msg and by alwadhapositionclustringPacket.msg, respectively. They contain node data parameters for use in the localisation algorithm, and they derived from the ApplicationPacket\_m class.

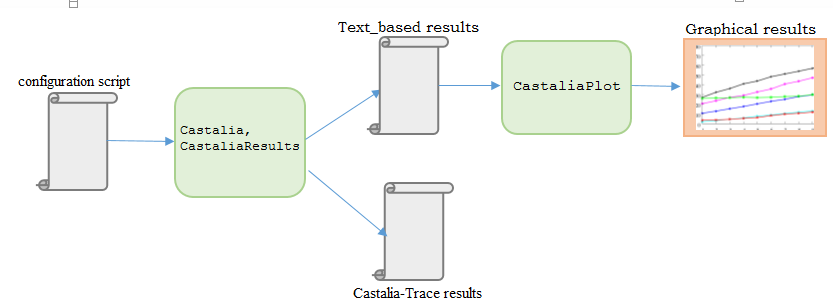
**5.3.1 The Structure of the New Castalia-3.2**

Figure 5.3 Shows the structure of the new version of Castalia, Castalia-3.2.

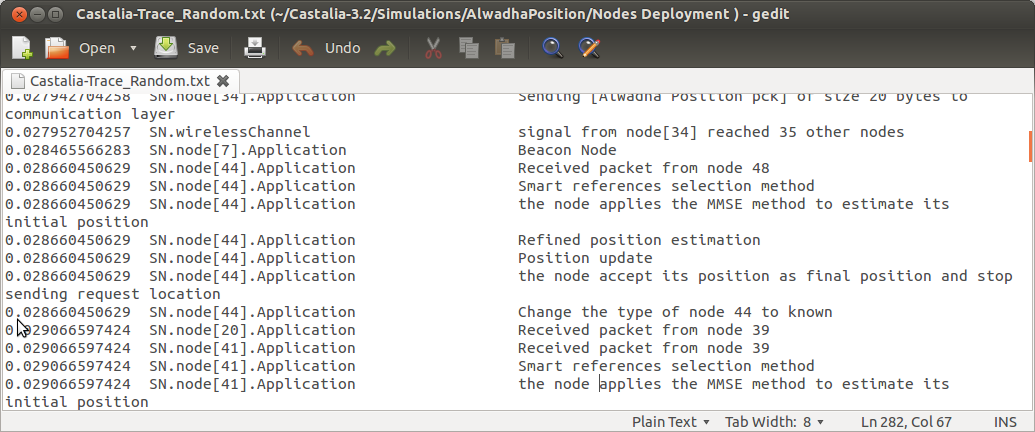
*Figure 5.3*. Structure of Castalia.

**5.3.2 Manipulation of the Output Files**

Castalia has three scripts to help with running simulations and interpreting the results, as shown in Figure 5.4: Castalia, CastaliaResults, and CastaliaPlot for creating graphs.

*Figure 5.4*. Tools used to manipulate the results.

In idioms of running the actual simulation, Castalia is a console-based application. There is a startup script in each application that calls the application module, which actually runs the simulation. Every sensor node has the capability to activate a trace. When the simulation is implemented, this trace information is kept in a separate text file, as in the example of the Alwadha algorithm in Figure 5.5, and stored in the application file directory

*Figure 5.5*. Trace file of Alwadha algorithm.

Castalia also offers a script that allows holding individual link qualities and storing this information in a separate text file. In addition, Castalia has a graph plotting tool called CastaliaResults, which we can use to measure different parameter values for individual nodes within the network. While running the simulation, there is CastaliaPlot, which generates a graph of the results. CastaliaPlot is designed to take the output of the CastaliaResults file as an input. CastaliaPlot uses gnuplot to produce the graphs.

**5.3.3** **Simulation and Evaluation**

This section evaluates the original Alwadha algorithm and the Alwadha clustering algorithm. The results show the effects of the nodes’ deployment, density, and network size on energy consumption.

**5.3.3.1 Environment**

The Alwadha localisation algorithm was evaluated using the WSN simulator (Castalia-3.2). The configuration file starts with the general section. The first line in the general section includes the command Castalia.ini, which assigns some parameters affecting general OMNeT execution as well as parameters that map the different RNGs to modules. It then defines the simulation time and the network parameters, such as the field size. The CC2420.txt file was used for the parameters of the real radio model, which are: dataRate(kbps):250, modulationType:PSK, bitsPerSymbol:4, bandwidth(MHz): 20, and noiseBandwidth(MHz):194.

In CC2420.txt, the eight possible transmission power levels were defined as follows:

Tx\_dBm 0 -1 -3 -5 -7 -10 -15 -25

Tx\_mW 57.42 55.18 50.69 46.2 42.24 36.3 32.67 29.04

The first line lists the output power of the different transmission levels in dBm, and the second line lists how much energy the radio is spending when transmitting at that power level. In addition, the number of nodes and the deployment type are listed.

**5.3.3.2 Results**

Several experiments were performed to evaluate the energy consumed when applying both the Alwadha localisation algorithm and the Alwadha localisation clustering algorithm, based on three factors: node deployment, node density, and network size. In each experiment, the results were evaluated, compared, and analysed to show the effect on the energy consumed by the network.

**5.3.3.3 Node Deployment**

Table 5.1 shows the complete setup environment of each experiment in node deployment factors. In the first experiment, the nodes were deployed randomly, and the second experiment utilised grid deployment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factor | Fig. | Deployment | Area | #Beacon | #Unknown |
| Node Deployment | 5.6 | Random | 200x200 | 12 | 52 |
| 5.7 | Grid | 200x200 | 12 | 52 |

Table 5.1: Node Deployment Environment

In this experiment, the nodes were deployed randomly. Figure 5.6 presents the energy consumption of the original localisation algorithm (AlwadhaPosition) and the Alwadha position clustering algorithm.

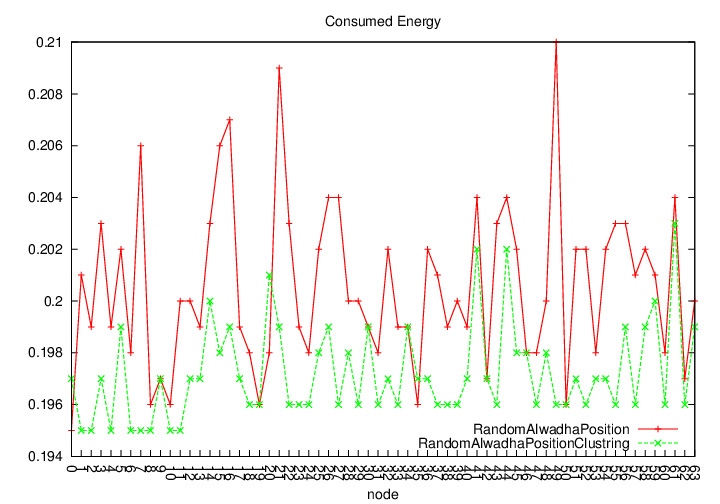
*Figure 5.6*. Random deployment.

Figure 5.6 shows the random Alwadha algorithm, which utilised a low number of references but increased the energy consumption of the WSN more than the other algorithm. With randomly deployed nodes, there can be many grids that have a small number of nodes. Thus, different cluster sizes will exist. In AlwadhaPositionClustering, CHs are distributed in an even way, leading to a decreased level of consumed energy.

In the second experiment, the nodes were deployed in the form of a grid network. Figure 5.7 presents the energy consumption of the AlwadhaPosition and AlwadhaPositionClustering algorithms.



*Figure 5.7*. Grid deployment.

As shown in the figure, the AlwadhaPosition algorithm used a low number of references, and energy consumption was decreased since the grid structure had the same number of neighbours. However, the AlwadhaPositionClustering algorithm achieved better results in terms of the amount of consumed energy since different clusters had the same size, meaning that the traffic load decreased and the time taken to select the CHs was better.

The results of the two previous experiments are summarised in Table 5.2.

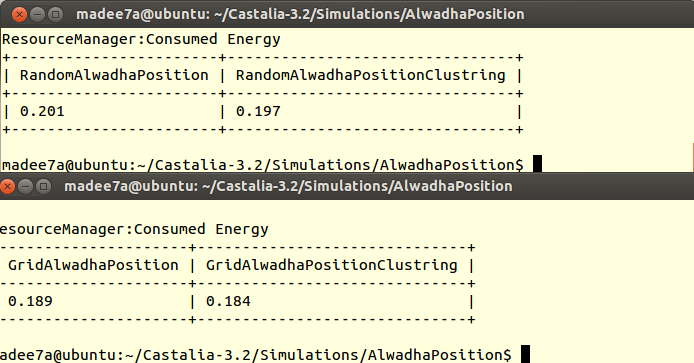


Table 5.2: Average of Energy Consumption of Node Deployment

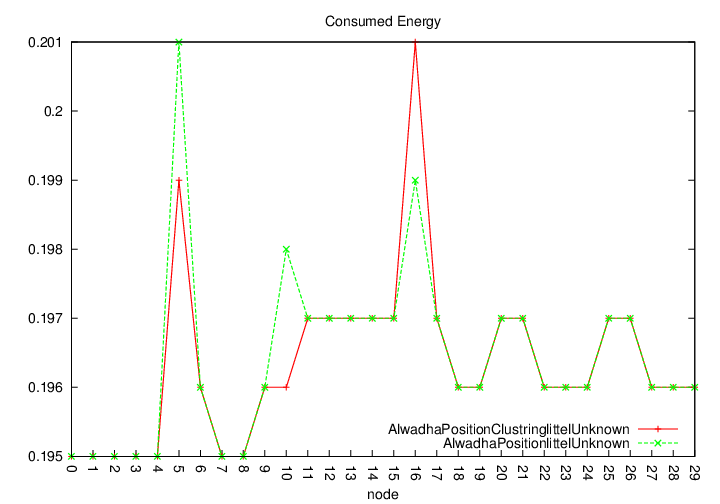
**5.3.3.4 Node Density**

The network consisted of two types of nodes: references and unknown nodes. This section examines the impact of changing the number of these nodes on the performance of the localisation algorithms. First, a fixed number of references was used while the number of unknown nodes was changed, and then the number of unknown nodes was fixed while the number of references was changed (Table 5.3).

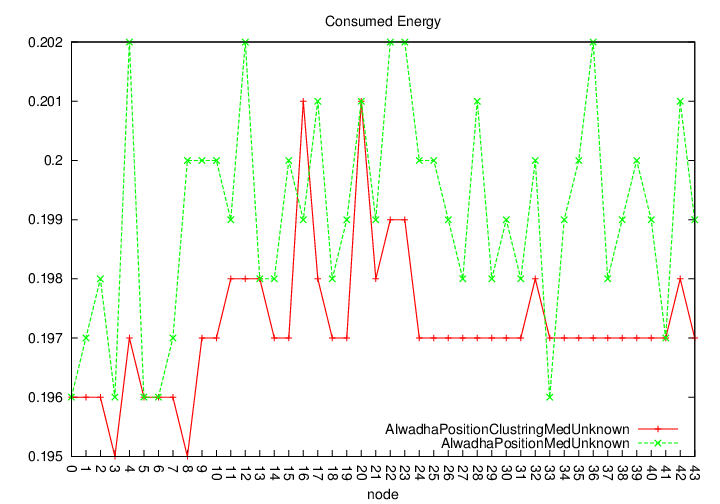
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factor | Fig | Deployment | Area | #Beacon | #Unknown |
| Node  density | 4.8 | Random | 200x200 | 9 | 21 |
| 4.9 | Random | 200x200 | 9 | 35 |
| 4.10 | Random | 200x200 | 9 | 51 |
| 4.11 | Random | 200x200 | 9 | 21 |
| 4.12 | Random | 200x200 | 23 | 21 |
| 4.13 | Random | 200x200 | 39 | 21 |

Table 5.3: Node Density Environment

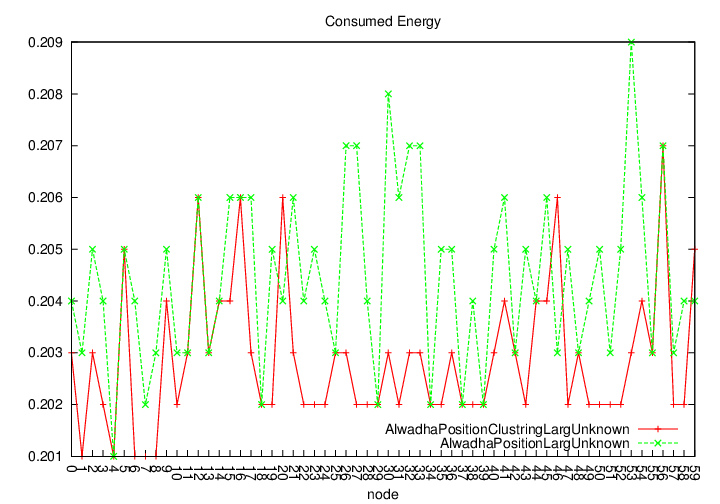
In the first three experiments, the nodes were deployed randomly. Figures 5.8, 5.9, and 5.10 present the energy consumption of the original localisation algorithm, AlwadhaPosition, and AlwadhaPositionClustering (Figure 5.8 a small number of unknown nodes, Figure 5.9 a medium number of unknown nodes, and Figure 5.10 a large number of unknown nodes). To examine the density of the unknown nodes, 9 beacons were used, with the density varying as 21, 35, and 51 unknown nodes.



*Figure 5.8*. Little unknown density.



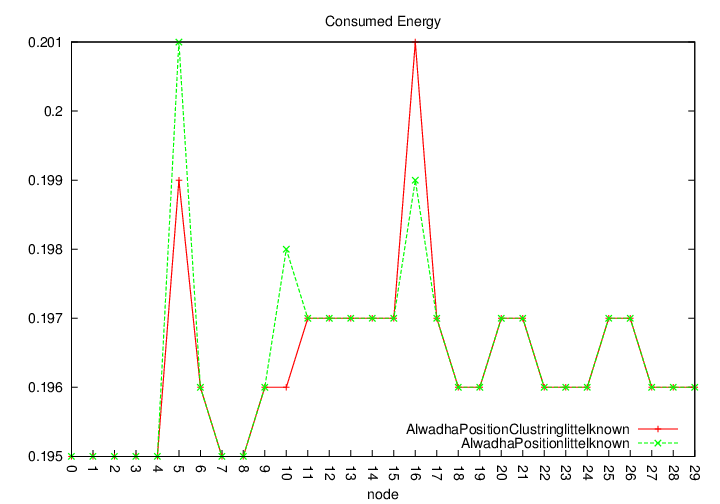
*Figure 5.9*. Medium unknown density.



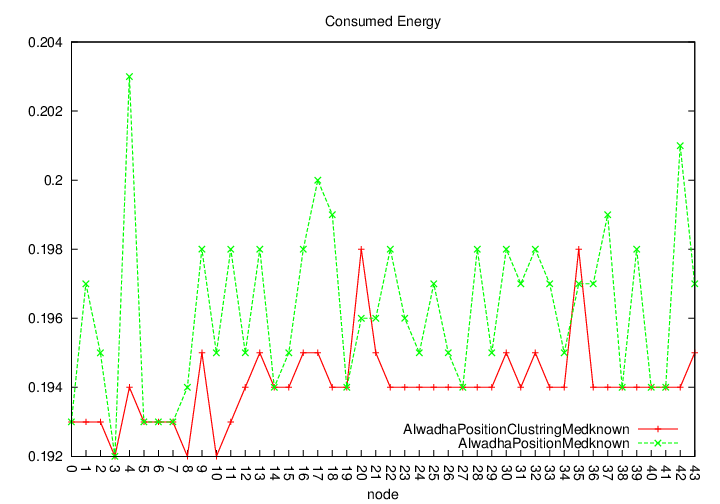
*Figure 5.10*. Large unknown density.

From Figures 5.8, 5.9, and 5.10, when using different densities of unknown nodes, the smart references selection method used by the Alwadha algorithm enabled the nodes to estimate their location with the best accuracy but also increased energy consumption compared to the Alwadha position clustering model. The energy consumption using the Alwadha position model increased as the traffic load increased due to increases in the density of the unknown nodes. With AlwadhaPositionClustering, energy consumption is kept low because it uses a CH selection decision, which results in a fair selection and even distribution of CHs, thus reducing energy consumption among all nodes in the network.

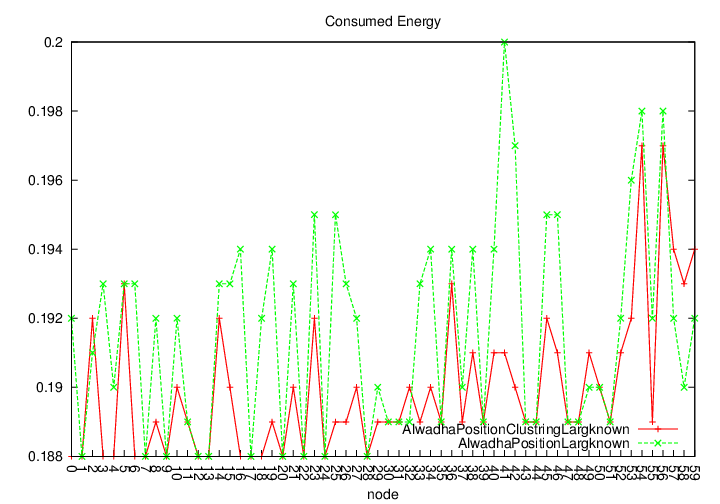
In the last three experiments, the nodes were deployed randomly. Figures 5.11, 5.12, and 5.13 present the energy consumption of the original localisation algorithm (AlwadhaPosition) and the AlwadhaPositionClustering algorithm (Figure 5.11 a small number of known nodes, Figure 5.12 a medium number of known nodes, and Fig 5.13 a large number of known nodes). To examine the known nodes’ density, 21 unknown nodes were used, and the densities varied as 9, 32, and 39 known nodes.



*Figure 5.11*. Little known density.



*Figure 5.12*. Medium known density.



*Figure 5.13*. Large known density.

From Figures 5.11, 5.12, and 5.13, when using different densities of known nodes, the Alwadha algorithm achieved the best accuracy. An interesting observation about Alwadha is that the average number of references used was reduced by increasing the density of the references. This is because the smart reference selection technique results in more references with a high probability of accuracy, which enables the nodes to select a smaller subset of references fulfilling the condition Pacc >= Pmin. With AlwadhaPositionClustering, energy consumption was lower due to the low overhead in the network because of the use of CH selection.

The results of these six experiments, which show the average energy consumption for each experiment using different densities of unknown nodes and references, are summarised in Table 5.4.

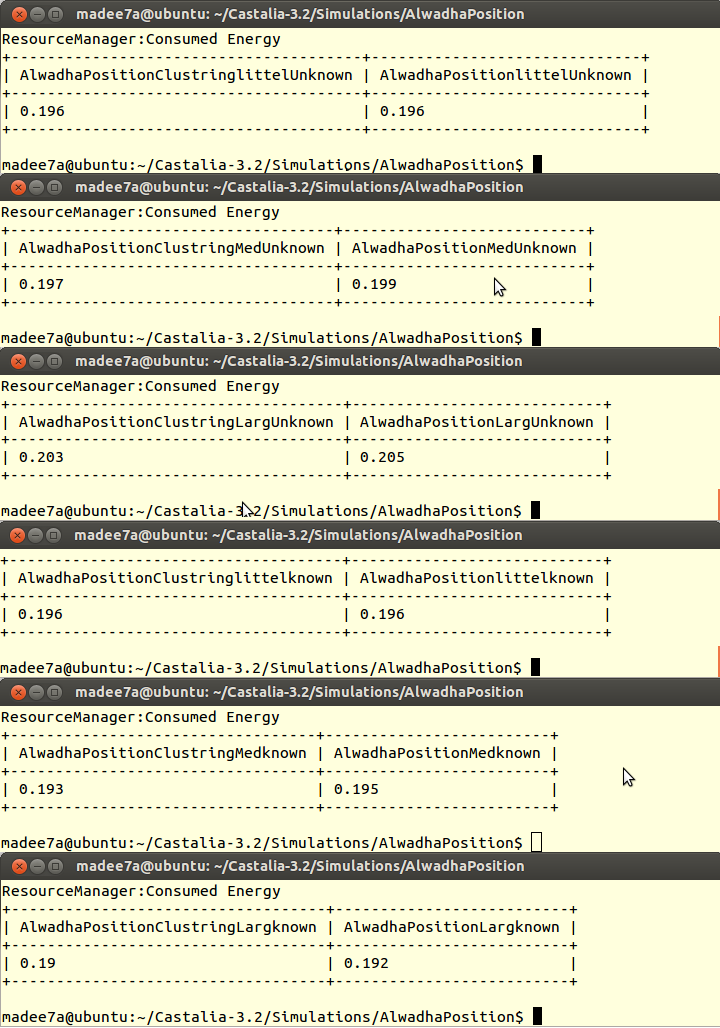


Table 5.4: Average of Energy Consumption of Node Density

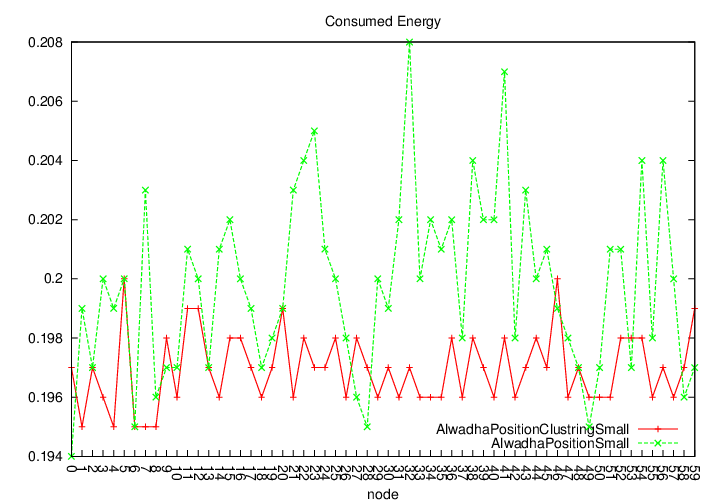
**5.3.3.5 Network Size**

To determine the impact of changing the network size on the performance of the localisation algorithm, three experiments were performed using different network sizes (small, normal, and large), as shown in Table 5.5.

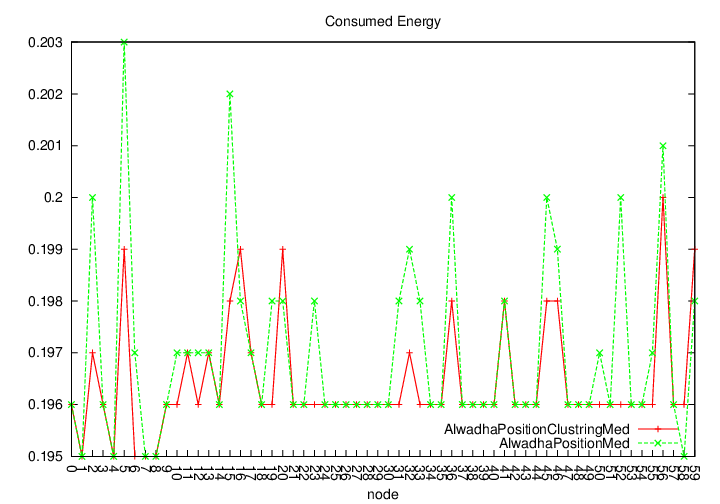
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factor | Fig | Deployment | Area | #Beacon | #Unknown | Notes |
| Network  size | 5.14 | Random | 100x100 | 9 | 51 | Small |
| 5.15 | Random | 200x200 | 9 | 51 | Normal |
| 5.16 | Random | 600x600 | 9 | 51 | Large |

Table 5.5: Network Size Environment

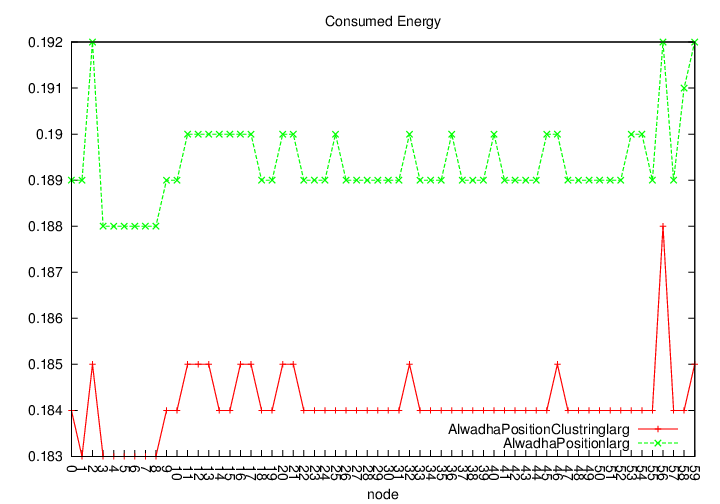
In these three experiments, the nodes were deployed randomly in differently sized networks. Figures 5.14, 5.15, and 5.16 present the energy consumption of the AlwadhaPosition model and the Alwadha position clustering model (Figure 5.14 a small network, Figure 5.15 a medium network, and Figure 5.16 a large network).



*Figure 5.14*. Small network size.



*Figure 5.15*. Medium network size.



*Figure 5.16*. Large network size.

From Figures 5.14, 5.15, and 5.16, when using different network sizes, the accuracy of the original Alwadha position algorithm increased and energy consumption decreased, while each node in the AlwadhaPositionClustering, after hearing CH advertisements, determined its closet CH on the basis of the received signal strengths of multiple CHs. This resulted in avoiding collisions and decreasing the energy consumed compared to the original algorithm.

The results of these three experiments, showing the average energy consumption for each experiment using different network sizes, are summarised in Table 5.6.

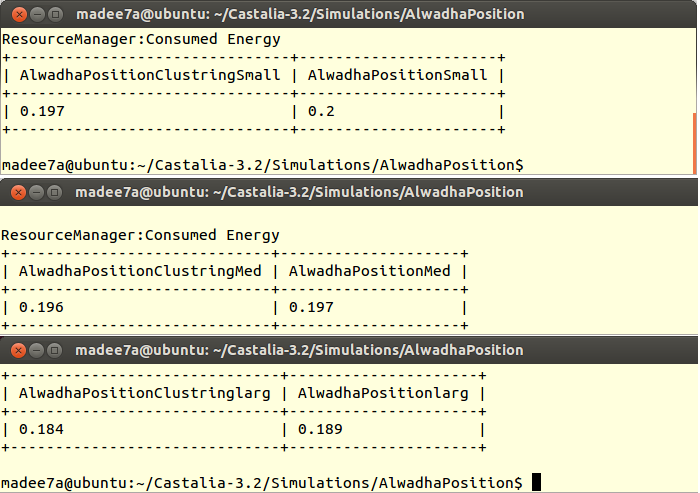
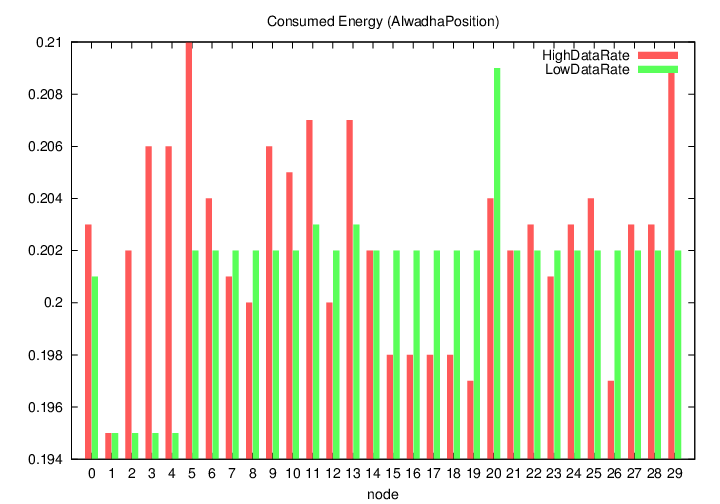


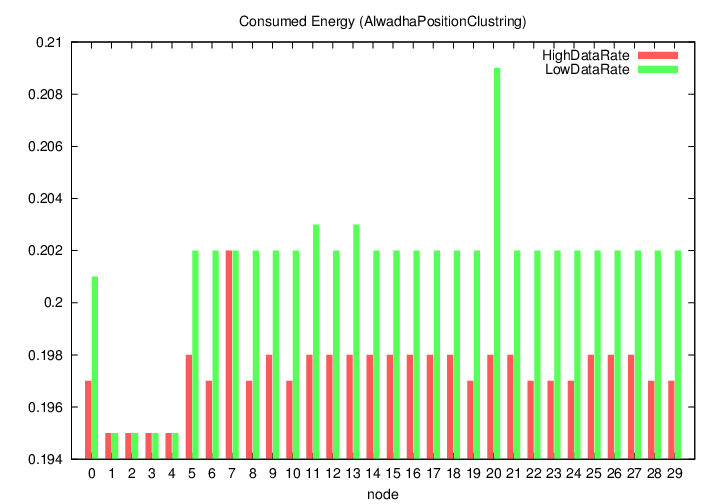
Table 5.6: Average of Energy Consumption of Network Size

**5.3.3.6 The Effect of Changing the Data Rate on Power Consumption**

This experiment was performed to evaluate the energy consumed using the Alwadha position algorithm and the Alwadha position clustering algorithm, based on different data rates: a low data rate (50 kbps) and a high data rate (130 kbps). The first figure below depends on the Alwadha position model and the second on the Alwadha position clustering model. In low data rate environments, nodes may have a lot of packets to transmit in most cycles. In such a case, the nodes waste energy during the data period. Approximately the same amount of energy was consumed using both algorithms in low data rate environments. However, with the high data rate there was a difference between the algorithms because the Alwadha position broadcasts to the whole network. The resulting high data rates require additional encoding, which requires additional radio traffic, resulting in increased power consumption by the radio. In Alwadha position clustering with the high data rate, the probability of broadcasting packets to the whole network and collision incidents is greatly reduced because of the short transmitting time to the nearest CHs.



*Figure*. Energy consumption of the Alwadha position algorithm.

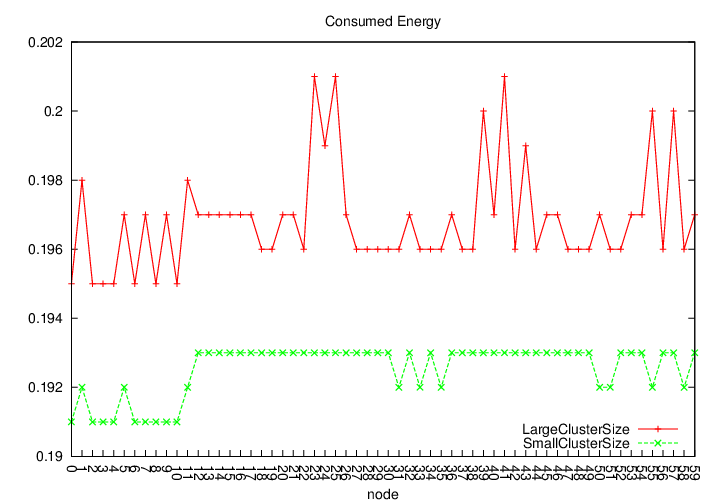


*Figure*. Energy consumption of the Alwadha position clustering algorithm.

**5.3.3.7 The Effect of Applying Alwadha Position Clustering on Network Performance:**

In this section, final thoughts about the proposed algorithm are offered.

* The accuracy of Alwadha position clustering is not affected because the mechanism does not change, but there is an effect on the performance of the WSN.
* The size of the clusters can affect the data aggregation process occurring at the CH nodes. When the data collected at the CHs are aggregated without summarising or compressing, this is called packet merging. In packet merging, the payload size requirement increases linearly with increases in cluster size, and hence the level of energy consumption increases.



*Figure 6.1*. Different cluster sizes.

* The Alwadha position clustering algorithm is only suitable for static nodes and does not work efficiently for mobility nodes.
* The performance of the WSN is affected when selecting the CH. The CHs consume memory constantly as a result of maintaining a table of their neighbourhood nodes. Updating this table consumes energy, as shown in Figure 5.10.
* Alwadha position clustering is understandable but requires considerable calculation to implement. It can take a long time to determine a solution, depending on the appropriateness of the reallocation criteria to the structure of the data, and marked by cubic time complexity (O(N³)).
* Nodes in the WSN consume time analysing the packet headers to decide whether or not they belong to them. This usually occurs when the CHs broadcast their residual energy levels and their grid IDs within their grid region.

**5.4 Conclusion**

At the beginning of this chapter, the Omnet++ simulator with Castalia 3.2 was presented and information was provided regarding how to tune this simulator for this study. Then, approaches for saving energy in sensor networks were compared: the original Alwadha position model and the Alwadha position clustering model were presented. The Alwadha position algorithm had good accuracy and utilised a low number of references, a smart reference-selection method, and a termination creation approach. This reduction in the number of references greatly improved computation, communication, and energy consumption. When clustering was added (as in AlwadhaPositionClustering), energy consumption further improved. This model results in the greatest energy savings when there is a traffic load, in large networks, or in grid deployment. Energy consumption resulted from the fair selection and fair distribution of the CH nodes. The remaining nodes, after hearing CH advertisements, determined their closest CH on the basis of the received signal strengths of multiple CHs and communicated only with that CH to avoidance collisions and overhearing. Alwadha position clustering improves the performance of the Alwadha position algorithm in WSNs.

**6 Conclusion**

**6.1 Conclusion**

WSNs have become a key technology and are increasingly applied to address processing and environmental problems. The major functions of sensor networks include localisation, tracking, navigation, and sensing. Sensor localisation in sensor networks is essential and critical because sensor position information is needed for positioning objects, tracking objects, targeting, routing in network, and in many other operations. In most situations, sensors are randomly deployed or are deployed in inaccessible areas. Thus, it is desirable to introduce robust and efficient algorithms that enable sensors to implement self-localisation. The best location of sensor nodes could result in decreased transportation energy costs and increased sensor network lifetimes. WSNs have finite energy availability, while sensor localisation generally includes energy-consuming calculations and communication. Therefore, it is necessary to decrease the energy needed for sensor localisation, and many approaches have been used for this purpose.

Most existing sensor localisation research attempts to provide accurate location estimation for a network of sensors. In this thesis, we chose to study and modify an accurate localisation algorithm called Alwadha [1], which selects the lowest possible number of references that is capable of providing accurate positions.

The aim of this thesis was to develop an algorithm for localisation to save energy in WSNs. Two methods were proposed for balancing energy consumption in a WSN.

The first, the new Alwadha position clustering algorithm, was proposed with the goal of increasing the accuracy of position estimation and saving energy in the WSN. The main concept of the Alwadha position clustering algorithm is to find the location of the sensor nodes in the network by first starting to search for their locations and then using a clustering algorithm to inform the sink as to their locations.

There are three stages needed to build the Alwadha position clustering algorithm. Initially, each node searches for its accurate location and then determines whether it accepts that location. In the second stage, if the node position is not accepted, the node sends its location to the sink using clustering terms or back again to search to its accurate location. Finally, to save energy, the nodes are divided into active and sleep nodes for each round of the algorithm’s execution.

The Alwadha position clustering algorithm uses two kinds of optimisation. First, the computational cost is decreased as a result of using a clustering algorithm that is responsible for sending node locations to the sink node. Decreasing the computational cost proportionally decreases the level of energy consumption. Second, nodes are divided into either active or sleep mode in each algorithm execution round.

We compared the two approaches in terms of the energy saved in WSNs. The Alwadha position algorithm demonstrated good accuracy and used a small number of references, a smart reference-selection method, and a termination creation approach. This reduction in the number of references greatly improved computation, communication, and energy consumption. When using Alwadha position clustering, energy consumption was further enhanced. The Alwadha position clustering model resulted in the greatest energy savings in the case of traffic load, in large networks, or in grid deployment because of the fair selection and even distribution of CHs, which reduced the consumption of energy among all the nodes in the network. The remaining nodes, after hearing CH advertisements, determined their closest CH on the basis of the received signal strengths of multiple CHs and then communicated only with that CH to avoid collisions and overhearing. In sum, Alwadha position clustering improves the performance of the Alwadha position algorithm in WSNs.

The second method for balancing energy consumption in a WSN is to use two of the MAC layer protocols associated with the original Alwadha algorithm: TMAC and TunableMAC. In TMAC, wireless sensor nodes are able to turn on their radios at synchronised times and turn them off after a certain time-out, when no communication has occurred for a certain period of time. TunableMAC is contention-based and uses a CDMA mechanism to mitigate packet collisions.

The Alwadha algorithm, the Alwadha algorithm under the TMAC model, and the Alwadha algorithm under the TunableMAC model were implemented, and the results were obtained to determine the effect on energy consumption levels of using different MAC protocols. Using the TunableMAC model, energy consumption was improved, but there were some problems (e.g. as traffic loads increased, energy consummation also increased). Moreover, TunableMAC’s fixed duty cycle increased energy consumption. However, it is a good model when a network is idle, and it provides the best energy consumption value. The TMAC model saved the most energy with a traffic load, in a large network, and with a high density of nodes. This is because TMAC employs numerous approaches that save energy, such as a dynamic duty cycle, control packet RTS, CTS, ACK, and FRTS to avoid collisions and overhearing. The Alwadha algorithm under the TMAC model resulted in more energy savings in our experiments than the other models.

**6.2 Future Work**

The importance of sensor networks is accepted by the research community, and many open issues need to be addressed. Defining the location of nodes in WSNs is a motivating and challenging research area characterised by many unanswered questions. The work presented in this thesis answers some of these questions while opening the door to many more new puzzles. Some future challenges and a probable follow-up of this work are summarised below.

*Mobile sensor networks:*

In this thesis, the Alwadha position clustering model was developed and investigated with static WSNs. An extension of this algorithm could be implemented by modifying and investigating the application to mobile sensor networks.

*Secure localisation system:*

Achieving a secure localisation system is another challenging area. Such a framework requires combining distance bounding into localisation projects. The distance bounding protocol that is used should follow the ‘principles of secure distance bounding’ defined in [21] and suitable for WSNs. Moreover, the performance must be evaluated, both in terms of overheads and resistance to attack. [1]

*Energy efficiency:*

As explained in this thesis, one of the most important challenges is improving the energy efficiency of the algorithms, and future work could use the LEACH protocol in the routing layer to achieve greater energy efficiency.

*Three-dimensional space:*

In this thesis, the localisation algorithm assumed that wireless sensor nodes are deployed in a planar (2-D) space. It would be interesting to research the algorithm for localisation in three-dimensional space.