Design Considerations for Fixed Node Assisted Multi-hop Mobile Sensor Networks.

By

Nadra Ben Otman

A Thesis Submitted in Partial Fulfillment
Of the Requirements for the Degree of

Master of Applied Science
In
Faculty of Engineering and Applied Science
And the Electrical and Computer Engineering Program

University of Ontario Institute of Technology
October, 2015

Supervised by:

Dr. Shahram Heydari and Dr. Ramiro Liscano

Abstract

Node mobility in Wireless Sensor Network poses a challenge to the routing protocol; it causes link breakages and disconnections between the nodes. This instability in the network leads to a drop in the successful transmission of data packets to the main station. In order to understand the key factors in the performance degradation in a mobile network and address them, a simulation based performance sensitivity analysis was done on a Collection Tree Protocol based network. First, the main reasons for packet drops in mobile networks were investigated. Then, the effect of the network size, node density and node speed is studied in more detail in a mixed mobile-static sensor network, as well as the effect of the number and transmission range of the static nodes in the network. Based on the performance sensitivity analysis, a set of criteria and network requirements is proposed, which can be used as network design suggestions for a mixed mobile-static sensor network to enhance the network's performance.

In Loving memory of my father

Dr. Abubaker Ben Otman

(February, 2015)

You were always the biggest driving force for my work and success.

To my dear Mother Gisela and my Sister Rania.

For all the love and support I always receive.

To my Husband Malek for the love, care, help and support.

And to my Children, Shaheen, Tamim and Reem who teach me every day not to give up.

Acknowledgments

I would like to deeply thank my supervisors Dr. Shahram Heydari and Dr. Ramiro Liscano for their patience, support and guidance throughout this work. I have learnt a lot from you; it was your thoughtful comments and suggestions that always made me look for more answers.

Thank you to my friends Amira, Jamie, Salma, Marta, Raihan and Rola for their help and support over the past years. And thank you to Gina who was always there when I needed.

Table of Contents

List of Figures

List of Tables

Introduction

1.1 Wireless Sensor Networks

Sensor nodes are little devices that can detect and measure changes in the physical environment and produce an output for it. The devices are usually restricted in power and memory; their task is mainly to sense the environment and send the data to a more powerful sink to process them. There are some more advanced nodes that are able to respond with an action on the surrounding, in this case they are called actuators.

Typically a sensor is built from the following components:

- Sensors: can be more than one in the same device, for example heat and pressure sensors.
- Power source: which is usually limited in resources, and it is not always possible to recharge the battery.
- Microcontroller: is responsible for the processing of the data and controlling other parts of the sensor.
- Transceiver: usually uses the RF frequency range or InfraRed. Important to know are the states of the radio (sending, receiving, idle) because they are defining the power consumption of the device. Depending on the device's, make there are different modes and a prefixed consumed amount of power for each of them; Sending mode, Receiving Mode, Idle Mode, and sleeping Mode are the most common ones. The three former ones use more energy, and it is usually best to keep the sensor in sleeping mode if nothing is being sent or received, switching from one state to the other as well consumes energy.
- Memory.

The usual sensor network consists of one/several sinks and a number of sensor nodes as shown in Figure 1.1. The sensor nodes sense the surrounding; the gathered data is then either sent periodically in applications that require continuous monitoring, or when a certain event occurs (a certain threshold is exceeded) and immediate attention is required. The sink is usually a device with higher computational power and larger memory; the sink node is able to process all the collected information and make decisions based on them.

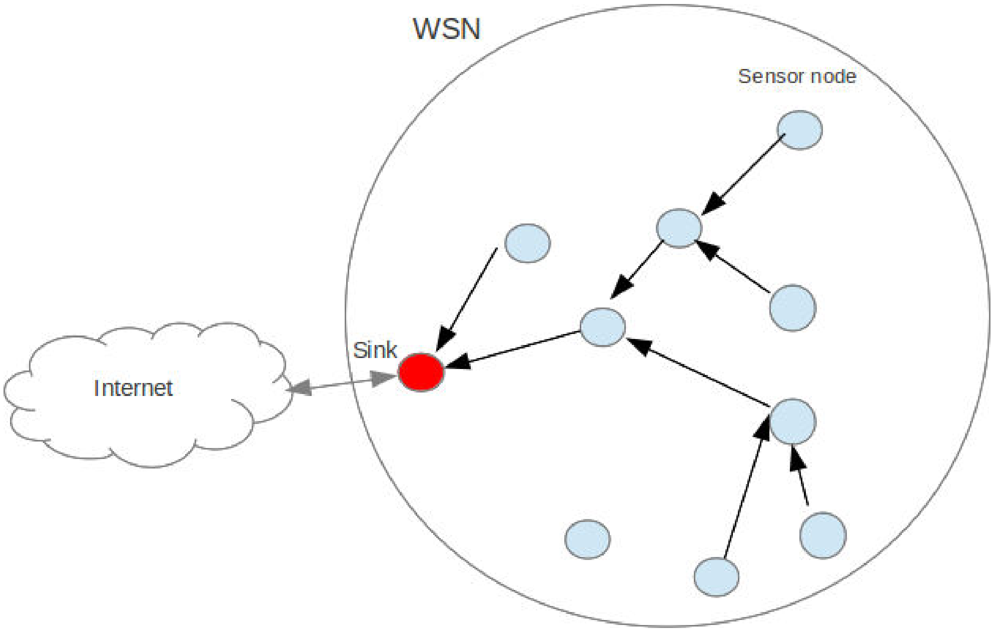


Figure 1.1: a wireless sensor network.

Based on the network topology the main two network types are:

- Single hop star networks, where all sensors are directly connected to the sink and send their data directly it.
- Multi-hop mesh networks, where there is a need to forward packets through other sensor nodes in order to reach the sink. Usually the nodes are connected to any node that is in their coverage area. This requires the network either to flood the packets until they eventually reach the sink or routing (directing) the packets in the network through specific chosen neighbor nodes until the packets reach the final destination (the sink).

Moreover, the sensor network can be part of a bigger network, where the sink is connected to another type of network, and the outer network has only access to the sensors through the sink. For example, in a home security system, the main board or the access point is connected wired and/or wirelessly to all the sensors, all of these sensors will send to the sink, and that sink is sending an alarm to the main station through the phone line.

1.2 Challenges of WSNs

The sensor nodes in the WSN are placed in the field in an ad-hoc manner; usually there is no pre-existing infrastructure and the nodes are decentralized and autonomous. The nodes connect to each other and try to send their data to the appointed sink. However, in building a WSN there is a number of constraints and challenges to keep in mind:

1.2.1 Energy

Sensor nodes operate on batteries, usually AA, flat or solar. In most cases, once the sensors are placed in the field they can't be recharged. Therefore, it is important to keep the energy consumption as low as possible to prolong the network's life time without compromising the performance.

To maximize network's life time and minimize power consumption several protocols have been implemented in the routing layer and in the MAC Layer. The authors in [1] propose AREA-MAC, where a linear optimization of the optimal duty cycle is presented to reduce energy consumption and latency. The node is put to sleep and keeps the power consumption very low for time T_1, then the node wakes up, samples, communicates and receives packets for time T_2. When the node is in the sleep mode it don't listen to the channel, send or receive data.

Other approaches are for example minimizing the control overhead; the trickle algorithm [2] was implemented to reduce the control packets. In a static network that becomes stable, in terms of paths and links, the nodes will gradually increase the time between subsequent control packets and hence reduce overhead.

1.2.2 Self Configuration

Certain applications require the addition of nodes into the network. The WSN should be able to easily accommodate new nodes without rebuilding the network or making any changes or modifications to the existing nodes in the network. The new nodes should be able to attach themselves to the network as soon as they are placed in it.

1.2.3 Fault Tolerant

Robustness is an important feature and challenge for the WSN, it means the network can handle node and link failures easily and can still perform well. In WSNs due to battery depletion for example, nodes die and links between nodes get broken. In this case, the network should have the ability to use other paths to reach the sink. Dekker and Colbert [3] argue that the best measure for network robustness is Network Connectivity, an indication of how well the nodes are interconnected. In other words having more than one path to the sink.

1.3 Wireless Sensor Networks and Mobility

Wireless sensing devices can be applied in a variety of applications. There is an increasing number of applications where some/all of the sensors or the sinks are mobile. The sensors can be static while the sink is moving in a predefined path to collect the information. Hence, less static nodes are deployed in the field. For example, in remote area monitoring the sensors are placed in different locations and have very limited communication with each other, so the mobile sink will move between the sensors to collect the information.

Habitat monitoring, such as Giraffe monitoring in the field, require sensors to be placed on the animals which are freely moving around in the field; in most cases there is no control on the direction or the location of the sensors. Making it difficult to follow them or keep connected at all times. Hence, it becomes a challenge to collect the data from all sensors in the fields.

1.3.1 Collection Tree Protocol

In multi-hop networks where the sensor nodes need to forward the data from neighbor nodes to the sink choosing an appropriate routing protocol becomes an important design factor for the network to deliver all data correctly and in time.

Due to the similarity between Mobile Ad-Hoc Network (MANETs) and WSNs, routing protocols that have been used in MANETs were applied in WSNs such as Ad-Hoc On-Demand Distance Vector AODV [4] and Dynamic Source Routing DSR [5]. However, there are differences between WSNs and MANETs and the devices used in both such as energy resources [6]. Usually MANET devices are larger, such as Laptops and PDAs, and usually these devices are around people and can be recharged when required. On the other hand, sensor nodes might not be recharged again once released into the field. Computational power is another limitation in sensor nodes, where usually the sensor nodes are more simple devices only used to interact with the environment and collect the required information to forward it. Finally, MANETs have one to one communication where devices interact amongst each other, while in a WSN the collected information is usually forwarded to the sink only.

Most sensor network applications don’t require data exchange between the sensors; they require the collection of information from the sensors towards the sink/sinks. Therefore a converge-cast type of routing protocol where all the sensors forward their data to sink/sinks is the best solution. The nodes don’t need addressing or information of all other sensors in the network. They just require information of their next neighbors. The Collection Tree Protocol CTP [7] is such a type of protocol. The sink advertises itself in the network through beacons (control packets). Each node that receives such a beacon calculates its link quality to the sink and advertises it. Nodes who receive beacons from neighbor nodes calculate the link quality to all of the neighbors they receive beacons from, if space is available in the limited routing table, then they choose the best next neighbor as a parent node to send their data to.

Mobility in the network causes frequent disconnections amongst the nodes, and it causes loops when nodes choose the wrong neighbor to forward their data to. Instead of forwarding data up the routing tree towards the sink, the packet gets retransmitted in a loop. The routing becomes challenging when there is no stable path to the sink. It becomes challenging as well to send non delay tolerant data to the sink. Mobility might as well cause the nodes to use more hops until the data reaches the sink which will cause time delay.

CTP has two important features that are helpful when dealing with mobility:

- The Pull flag: outgoing packets or control beacons will have a pull flag set to 1 if there is no path to the sink or if the link to the parent node gets disconnected.
- Loop detection mechanism: when a node discovers that its current parent might be a child of its own in the routing tree, it will request a new path to the sink and it will change the current parent.

However, even with the loop detection mechanism and the pull for path flag implemented, CTP doesn't perform well in mobile Networks [8]. The Packet Reception Ratio (PRR) falls below 20%, with high overhead in terms of packet retransmissions in the network. The performance analysis showed that CTP in its current form is not suitable in mobile WSNs.

1.4 Problem Statement

Mobility introduces link breakages to the WSN, which causes degradation in the performance of the network such as the PRR. The quick changing topology makes it often hard for the network to quickly react to the changes. In order to cope with the changes, in a multi-hop network, the routing protocol used in the network has to be able to quickly react to broken links and repair only the part of the network that is down, and it has to easily identify loops occurring in the network. Moreover, it has to proactively initiate a route request from neighbor nodes using the pull flag if no path is available.

The goal of this research is to propose a set of minimum requirements for CTP-based networks, in order to maximize their performance in mobile sensor networks. We chose CTP because it was designed for WSNs, and takes into consideration the limitations of sensor nodes. Moreover, out interest is in a collector style protocol. Our research uses Fixed Node Assisted-CTP (FNA-CTP) that was proposed by Sharma et al [8]; FNA-CTP has shown that it increases the performance of the network with mobile nodes without increasing control overhead. The protocol uses a number of static nodes in the network with larger coverage area and transmission power to cover most of the network. The mobile nodes can use the static nodes as backups when they can't forward their packets to a mobile node.

1.5 Objectives

This work aims to build on top and extend the previous work by Sharma et al [8]; the previous work focused on the performance of CTP in mobile scenarios and proposed introducing fixed nodes to improve performance. This work will focus on the following:

- First explain in more detail the protocol's advantages and shortcomings in mixed mobile-static networks using simulation based performance sensitivity analysis.
- Outline the network parameters that are important in dealing with link breakages due to mobility.
- Extend the work in [8] to different network sizes and a different number of nodes.

- Investigate the effect of node speed and transmission power on the network's performance.
- Propose a set of tables that provide network requirements to achieve the best PRR in terms of the number of static nodes and their transmission range.
- Vice versa, show the number of mobile nodes that can be introduced to the network for a required PRR.

1.6 Methodology

First, a literature review of related works and backgrounds was performed to investigate existing routing protocols that are used in mobile WSN applications; their performance and their key features of dealing with node mobility. Moreover, related works in graph theory, its measure of connectivity and its relation to mobile WSNs were reviewed.

Then, simulation based performance sensitivity analysis was performed in Castalia [8] on mobile networks running CTP and FNA-CTP. Different scenarios were investigated by varying the network size, number of nodes, number of static nodes, node speeds and transmission range of static nodes. In applications of larger scale networks with bigger field size or a bigger number of nodes it is usually easier to first test the network in a simulation setup to investigate the feasibility and advantages of the suggested protocol before investing into a real implementation. The performance results of the different scenarios have been used to provide suggestions and tables that can be used as guidelines in designing a WSN with mobile nodes using FNA-CTP.

1.7 Thesis outline

The thesis is organized as follows: Chapter two discusses related works and what has been done in the field of mobile WSNs. Chapter three first explains the collection tree protocol in detail; it will as well provide a list of the important parameters that effect mobility handling in the network. A full description of the

Fixed Node Assisted-Collection Tree Protocol is in section 3.5. Chapter four describes the simulator used, the simulation scenarios and the parameters in each of them. Chapter five discusses the advantages of having mixed mobile-static nodes for the performance of the network, and it discusses in more detail the parameters that are further important in mobile scenarios. Chapter six provides the simulation results for the different network scenarios when varying the number of nodes, the node speed, the number of static nodes and their transmission range. First, the results of all mobile network scenarios and then the results of the mixed static-mobile networks are provided. Section 6.4 will provide results analysis and discussions. Chapter seven provides a set of tables that can be used as general guidelines for designing a network when using FNA-CTP in mobile networks. Finally, chapter eight will conclude the work and provide future works related to this research.

Chapter Two

Related Works

This chapter provides a summary of other works done related to this work, it provides as well other reference works that have been used to progress with this research. The chapter is divided into three parts; section 2.1 will provide an overview of the routing protocols that have been applied in mobile WSNs and the different approaches taken. Then, section 2.2 provides a short overview of the validity of simulation based performance analysis and the validity of mobility models in simulation environments. Finally, section 2.3 describes related works in network connectivity in graph theory that will be used in the network's analysis.

2.1 Routing Protocols.

The starting point to this work was to understand the challenges that a network faces when nodes are mobile. Gerla et al [9] explain the problems associated with mobility in Ad-Hoc networks MANETs, and although our work is in mobile WSNs the problems discussed in Gerla's work in MANETs apply in WSNs as well. The authors point out the key challenges facing mobile networks. Most important challenge is the link breakage between the nodes. Mobility, especially when the nodes move with very high speeds, causes very frequent link connections and disconnections. Connection time is as well associated with link breakages. When two nodes move into each other's communication range with high speed they might be only hearing each other for a very limited time; this is not enough to establish a connection and use each other as a possible next hop towards the sink. The authors propose some techniques to handle the disconnections such as link predictions and the need for backup paths. This can be applied for example in those scenarios when the mobility pattern can be anticipated, or in scenarios where some of the nodes have a higher percentage to keep close to each other for longer periods of time due to their moving pattern. Another technique the authors proposed is using GEO Routing, where the nodes carry a GPS and know their own location; when they send their data they include their location so the neighbor nodes can extract location information and use it. Overhead due to frequent updates is another problem facing mobile networks as well as long disconnections due to node partitions in the network; the mobility pattern causes the nodes to partition into several disconnected groups. There are other solutions provided but due to the different nature of MANETs can't be applied as is in WSNs.

The routing of the packets through the network is most affected by the mobility. A node can't successfully forward the packets to the sink if there is no consistent valid path to the sink. To deal with this challenge several approaches are made. First one is to use existing MANET routing protocols and apply them in mobile wireless sensor networks such as AODV and DSR. However, many WSN applications only require the collection of information from the sensor nodes in the network using each other as next hops towards the sink; there is no need for an extra exchange of communication between the nodes, as it will only use more energy and hence cause faster battery depletion. Jambli et al [10] investigated the performance of AODV in mobile wireless sensor networks. The authors study the effect of the topology changes due to node mobility on the PRR and on the energy consumption of the network through simulations. Their results showed that there is a high packet loss in the network and an increase in energy consumption when AODV is used. The authors evaluated as well the performance of AODV under different node speeds, and their results showed that the PRR of the network decreases with the increase in node speed. Moreover, the authors investigated the effect of the number of mobile nodes vs. the total number of nodes in the network on the PRR; they simulated a small network with 10 nodes and varied the number of mobile nodes in increments from a full static to a full mobile network with a fixed sink. Their results revealed that with the increase in the number of mobile nodes there is a decrease in PRR which they related to the increasing number of broken links and the limited time the nodes have to update their routing tables to keep up with the broken links. They finally conclude that AODV is not able to react to topology changes and broken links to perform well in mobile WSNs.

Other researchers designed routing protocols specifically for mobile WSNs. However, these protocols usually lack real time implementations and standardizations. An example of such a protocol is the work of Shiny et al [11], they propose an Ad-Hoc on demand multi path routing protocol that finds multiple paths to transfer information from the sensor nodes to the static sink; the nodes advertise their current hop count to the sink, and the source nodes will use an alternate route if the current route is not working.

Another approach is to use existing protocols and apply them in mobile networks and see if they can be modified to be able to handle link breakages and frequent topology changes. One example is the Low Energy Adaptive Clustering Hierarchy LEACH [12] routing protocol designed specifically for WSNs, it is a hierarchical clustering routing protocol where the nodes choose nodes amongst them as cluster heads to forward their data to. LEACH-Mobile [13] protocol supports mobility in wireless sensor networks; each sensor uses a two way communication mechanism to become part of a cluster. The cluster head sends a message to the sensor nodes in its cluster and if it does not hear from a sensor node it is assumed to have moved out of the range of the cluster. When a node does not hear from the cluster head, it tries to connect to other cluster heads. However, this protocol suffers from a high number of packet losses and high energy consumption due to the overhead of the cluster membership management mechanisms.

Sharma et al [8] focused on CTP in their work, because it was designed for WSNs and it takes into consideration the sensor node limitations in power and storage.

Moreover, previous work showed that it performed well in static WSNs [14]. Their goal is to use CTP and modify and enhance it to accommodate node mobility in networks. In [8] and [38] a performance evaluation of CTP in mobile scenarios was performed; showing that the PRR of CTP is dropping significantly, concluding that regular CTP doesn't perform well in mobile scenarios. In [38] a comparison between AODV and CTP was conducted in mobile scenarios and their results showed that even though CTP was designed for static networks, it outperformed AODV in terms of PRR and control overhead. The authors proposed FNA-CTP as an enhancement to CTP; a number of fixed nodes are introduced to the network to act as backup nodes in case of link breakages. The fixed nodes can have higher power (larger transmission range) and a bigger buffer size than the mobile nodes. In their work, the fixed nodes advertise themselves as fixed nodes and cover almost the whole network, and if a mobile node receives a beacon from a fixed node it will add it as a special neighbor in its routing table. Once a node reaches the maximum number of retransmissions to its lost mobile parent it will search in its routing table for a static node to retransmit the packet to. The results in [8] show an improvement in the network's performance in terms of PRR and control overhead when compared to standard CTP.

A similar collection protocol is the Routing Protocol for Low Power and Lossy Networks RPL. Although RPL and CTP share the same basics, RPL is more complex than CTP. Both are collector-style protocols, but CTP is considered one way transmission from the sensor nodes to the sink while RPL can be two way. Moreover, both protocols make use of the trickle algorithm to reduce control overhead. However, RPL can use different metrics to build different trees in the network simultaneously, such as delay, cost and number of hops. The advantage of RPL is that it is IPv6 compatible, which means for applications such as smart home monitoring an easier connection into the sensor network through the internet and from outside the network. Similar to CTP, RPL was designed for static networks. Nonetheless there is interest in applying it into mobile networks. In [15] the authors propose a Mobility Enhanced-RPL, their assumption for the network is that the network has a mix of fixed and mobile nodes in it, and is not fully mobile. The mobile nodes advertise themselves as mobile nodes. Not like FNA-CTP, where the mobile nodes are not forced to use a fixed node as a parent in the Mobility Enhanced-RPL the nodes are forced to choose a fixed node as their parent if they have a fixed node in their routing table. Moreover, there is a change in the speed of solicitation messages to be able to handle the frequent topology changes. Same as in FNA-CTP they propose to remove the trickle in sending control messages and instead send control packets at regular more frequent intervals to be able to quickly react to link changes.

Ko et al [16] provided a performance comparison between CTP and RPL in static networks, and their results showed that RPL had a similar performance in PRR to CTP. However, the results in Radoi et al [17] showed that the received PRR in the network running CTP was higher than the network running RPL. Comparing the protocols they concluded that CTP is fault tolerant while RPL is not.

Le et al [18] investigated applying RPL in vehicular networks. The authors made certain modifications in RPL to be able to handle node mobility. They proposed turning off the trickle, and replaced it with an immediate ETX request for a newly discovered neighbor, to be able to quickly switch to a better parent if available. Furthermore, they introduced a loop detection technique to avoid sending to a parent that is a child of the node or its children, which can happen frequently in a mobile network. The authors showed that their proposed modifications enhance the performance of RPL in mobile networks, stating that RPL doesn’t adapt to node mobility in its regular form. Moreover, they showed that speed has an effect on the connection time between two nodes, when nodes move slower they stay connected for a longer period of time.

Reinhardt et al [22] proposed a CTP based routing protocol that is able to perform well in mobile networks. Their approach is to introduce a bloom filter into the node, where the neighbor nodes, and the possible descendant of a node are included. They used CTP as the underlying routing protocol. The goal is to efficiently save the information of the neighbor nodes without using too much space. Moreover, they introduced a gradually forgetting feature to the filter for mobile networks, where the node can eliminate the outdated nodes from its routing table to be able to keep up with the frequent topology updates. The authors emphasized on the importance of a fast updating routing tables that eliminate older nodes from the routing table to have space for the newer available nodes. They proposed a decrease in the upper bound for the trickle algorithm to be able to better react to the topology changes. The idea was to replace the dynamically growing routing tables at each node, and have a dynamic supported memory allocation. However, the authors tested the proposed protocol in a small sized network with only 10 nodes and two mobile nodes moving with low speeds. They didn't extend their work on larger networks with higher speeds and more mobile nodes. Moreover, their focus was more on extending CTP to enable point to point communication.

2.2 WSN network simulators and mobility patterns

In order to investigate a proposed or an existing protocol there are different methods that can be used. The most extensive yet reliable method is to build a real time network, apply the protocol that needs to be studied and try different scenarios to gather data for performance analysis. However, this approach can be very expensive in time and cost. The easier approach is to first analyze the protocol in theory and then implement it in a network simulator to study its performance. This approach makes it easy to try different scenarios and situations without extra cost. There is a number of surveys that list available WSN simulators including their key features and limitations [23, 24]. There has been research as well on the validity of such simulators and the correctness of their results compared to real time implementations [25]. There are several WSN simulators available that can be used in research such as Omnet++ [27] based MiXiM [28] and Castalia [29], QualNet [26] and NS2 [30]. Castalia makes it easy to extend any protocol and make modifications or add new protocols. Moreover, the simulator includes realistic radio and channel models. It supports node mobility, including the option to implement specific mobility patterns for certain applications.

The mobility model used in the simulator is as well an important factor when studying a protocol in a mobile environment. It is important to have mobility models for different applications based on realistic node behavior, in order to realistically simulate how the nodes move and investigate the network's performance. For example, Nardis et al [19] proposed group mobility model (DynaMo) that is able to simulate soccer players in the field playing in a group as well as individual mobility pattern. They show that their mobility model resembles the real mobility pattern of soccer players in the field. This mobility model was then used by Garcia et al [20] to investigate the performance of body area networks applied on soccer players for health monitoring. Their focus is on the collection of data from the soccer players during a game. And because usually the soccer players move together following the balls direction, they used a specific implemented group mobility model that applies to soccer players. The authors used AODV, a multi-hop routing protocol, where the players of both teams can be used as next hops towards the sink. Their goal was not to use a more general mobility model, because they wanted to investigate a protocol in a specific application where the mobility model of the nodes was known.

Dhamdhere et al [21] implemented a real time experiment for the soccer player's health monitoring. Each of the players had a mote and a GPS on their arms. The goal of this research was to find a suitable routing protocol that performs well in a soccer game. They investigated different available routing schemes and studied their performance mostly from the delay point of view, and they tried to minimize the time delay of sending packets to the sink. The authors found direct transmission to be the worst protocol in terms of delay, stating that the characteristics of the operational environment caused high delay. They concluded that multi-hop had lower time delay than single hop routing. FNA-CTP is as well using a multi-hop scenario as opposed to the single hop scenario and the results in [21] emphasized the use of multi-hop scenarios, which is similar to the findings of [20].

One of the most used mobility models is the Random Way point mobility model [32]. The nodes randomly choose a destination point and move towards it for a given time period and then stop for another time period. After the stop the nodes will change their direction. The mobility model will be discussed in more detail in chapter four.

2.3 Connectivity

In order to be able to receive all packets from the sensor nodes there should be at least one path per node towards the sink. There has been studies about network connectedness in MANETs as well as in stationary networks. The goal is to have a fully connected network where no nodes are isolated or isolated clusters are formed. Ideally, this means that the minimum node degree (the number of neighbor nodes) in the network should be 1 to ensure that no node is isolated. However, when each of the nodes is connected to only to one neighbor redundancy and robustness becomes an issue in case of nodes that die or loose connectivity. Xue and Kumar [33] proposed an average node degree for the static network to ensure a fully asymptotically connected network which is calculated by

$$Minimum\ node\ degree = 5.1774 \times Log\ n \qquad Eq.1$$

Where n is the number of nodes in the network.

Bettsetter [34] provided observations in regards to uniformly distributed nodes in the field. The nodes that are placed at the edges of the network will only have links towards the middle of the network; hence, their node degree is usually lower than the nodes in the middle of the network. This makes it difficult to compare the theoretical results with the simulation results. However, the author suggested taking a square in the middle of the network, and then to count all the outgoing links for a more realistic node degree calculation. This is helpful in the calculations of larger networks where it is easier to take only a part of the network to calculate the node degree. For example, in the vehicular networks only parts of the streets are used for simulations as it is impossible to implement the whole map.

Another look at network connectivity is to relate it to three important factors: number of nodes, network size and the transmission range of the nodes. For example, Santi and Blough [35] worked on the critical transmission range to ensure a connected network. They found that for a two dimensional network of size d^2, a number of nodes n and a node transmission range R, the relation to ensure a connected stationary network would be

$$R^2 n = d^2 Log_2 d \qquad Eq.2$$

However, in a mobile network either the number of nodes or the transmission range should be higher. It is suggested using eq.2 as a lower bound. The authors simulated a mobile network with two different mobility models. They concluded that network connectedness is affected by the number of mobile nodes in the network (the percentage of the mobile nodes with respect to the total number of nodes) and not by the mobility model itself. Moreover, the authors concluded that if it is not important to ensure 100% connectedness in the network at all times, energy can be saved by having a lower transmission range. For example, a network 150 by 150 meters, with 108 nodes then the minimum transmission range that the node should have is 38 meters to ensure that network connectedness with high probability. If the transmission range for the nodes is known, for example having nodes with a transmission range of 29 meters then we need a minimum of 193 nodes to ensure a connected network.

However, it is important to distinguish between connectedness and the reception of all application packets from the nodes. Links going up and down frequently and quick topology changes make it difficult to forward all the required data to the sink. Even if the network is fully connected, it will take the routing protocol some time to update broken links. Another factor is the interference and the collisions at the radio level and at the MAC layer especially with higher node densities.

Chapter Three

CTP and FNA-CTP

In this chapter, sections 3.1 to 3.4 will provide a detailed description of CTP, its main components and key features. Section 3.5 discusses FNA-CTP. In order to apply any protocol in a different scenario it is important to completely understand how the protocol is performing and its execution. In this research, the goal is to apply CTP in mobile scenarios and using it in FNA-CTP; this makes it important to know the key features that help increase or decrease the performance of CTP in mobile scenarios.

3.1 Collection Tree Protocol CTP

CTP is an address free routing protocol that aims to collect data from several nodes and forwards them to the sink. The protocol is a converge-cast protocol; several nodes send their information towards one single sink node. However, it is possible as well to have several sink nodes or several trees towards one sink based on different criteria.

The nodes send their data to the sink; they choose the next best neighbor to become the parent node, to forward their data to, until they reach the sink. In CTP, choosing the best neighbor, as a next hop towards the sink, is based on a link gradient called ETX (Expected number of Transmissions). Each node will calculate a 1-Hop ETX value to all its available neighbor nodes. Then, the overall ETX is calculated by adding the 1-Hop ETX values of all parent nodes. Section 3.2.2.1 describes in more detail the ETX calculation.

3.2 Basic CTP Components

First it is important to understand the different components of CTP in each node:

3.2.1 The Routing Engine

The routing engine RE is responsible for the sending and receiving of beacons. It is responsible as well for the frequency of sending beacons. The beacons in CTP are sent using a trickle algorithm; this means that, in the set-up phase, the beacons are sent out in smaller time intervals, then the time interval is doubled by each successive transmission until reaching a maximum pre-set interval. The main reason for implementing the trickle algorithm in CTP is the reduction of control traffic. The trickle algorithm [2] has proven to reduce control overhead in static networks.

However, there are some cases when the beacon interval is reset to the minimum beacon interval value. For example, when a loop is detected, the node will request a path by enabling the pull flag, and then all the neighbor nodes overhearing the request will reset their beacon interval.

The RE has also to build and update the routing table in each node; it will hold the information about the neighbor nodes and their overall cost to the sink (multi-hop ETX). For example, the ETX to the sink through three neighbor nodes can be 16 (6+5+5) or through four other nodes 12 (3+4+2+5). Moreover, the RE is responsible for choosing a parent and replacing the parent when needed [37], as in the previous example, the node with the ETX of 12 will be chosen as a parent.

3.2.2 The Link Estimator

This component is responsible of building and updating the neighbor table, which holds the information about the current neighbors of a node and the 1-Hop ETX value to each of them. By default, CTP's neighbor and routing tables have space for

10 neighbors. Both tables have the same entries and are related to each other. However, a neighbor will not be available until the 1-Hop ETX value is calculated [37].

The details of how neighbors are inserted and the 1-Hop ETX is calculated are important to understand how these values are updated.

3.2.2.1 The 1-Hop ETX calculation

The 1-Hop ETX value is calculated based on the node's outgoing or incoming link quality. The outgoing quality is calculated using the number of successful transmitted unicast data packets to the node's parent. The incoming link is calculated from the number of beacons received in a pre-defined time window.

3.2.2.1.1 Calculation of the incoming link quality

The node has to calculate the 1-Hop ETX based on the incoming link quality in the following cases: in the startup phase of the network, or when a node has just joined the network with an empty neighbor table, or if there are no application packets to send. The calculations are explained in detail in [37].

When a node sends a beacon, it will include the current parent and the overall ETX value to the sink, this value will be used in the routing table. The link estimator will attach the sequence number of the beacon as a header to the outgoing routing packets. If a node hears a beacon from a new neighbor, it will see if it has space in its neighbor table to inserts it. Then, it starts counting the number of beacons received from that node(n_b). The beacons include the sequence number of each beacon and the total number of transmitted beacons by that node (N_b). The quality of the incoming link is calculated as follows:

$$Q_b = \frac{n_b}{N_b} \qquad \textit{Eq.3}$$

This value is calculated over a pre-fixed default time window(w_b). Every w_b the outgoing link quality value (Q_b) has to be updated. Then, the value Q_b is passed through a weighting filter to average the current and previous samples.

3.2.2.1.2 Calculations of the outgoing link

If the node already has a parent and application packets to send, the outgoing link can be used to update the 1-Hop ETX value [37].

Similar to the incoming link quality calculations, the number of successful transmissions is counted for a pre-defined default window of time(w_u). If the number of unicast application packets, including retransmissions, sent to the parent is(n_u), and the number of acknowledgment packets from the parent is(n_a), then the quality of the outgoing link is calculated by:

$$Q_u = n_u/n_a \qquad \textit{Eq.4}$$

The value Q_u is reset after(w_u) and it is passed through a weighting filter.

Finally the value of the 1-Hop ETX is then calculated as follows:

$$ETX_{1Hop} = \alpha_{ETX}Q + (1 - \alpha_{ETX})ETX_{1Hop}^{old} \qquad \textit{Eq.5}$$

Where Q can be either Q_u or Q_b whatever value is available. And α_{ETX} has a default value of 0.9. Depending on which of these values is updated more frequently the 1-ETX will be more frequently updated. For example, if the beacons are sent every 250ms and the nodes send application packet every 3ms, then the 1-Hop ETX value is updated based on the application packets.

In case of mobile nodes, links are disconnected frequently. The large spacing between beacons makes it difficult to quickly update the 1-Hop ETX to keep up with the frequent link disconnections.

3.2.3 The Forwarding Engine

This Forwarding Engine FE is responsible for sending application data packets, either the node's own packets or packets received by its child nodes. The FE is as well responsible of detecting loops and duplicate packets to discard them early.

3.3 CTP Tree Creation

In order to be able to use CTP in mobile scenarios, it is of advantage to understand the exact way CTP is building and maintaining the routing tree; the way the protocol reacts to link breakages and disconnections from the parent node. The exact details are explained in [37].

In the CTP tree creation phase, the sink initiates the tree by broadcasting beacons with the ETX of 0. If there are multiple Sinks in the network there is an identifier tag in the beacon to distinguish between them. If a node hears a beacon for the first time, regardless of the origin being the sink or another node, it will search for an empty space in its neighbor table for the new neighbor. If the table is full, it will check if it can evict one of the neighbors; one that is not a current parent and that hasn't been updated for a default time window. If the beacon is from the sink node it must be inserted into the table even if it is full. Once a new neighbor is inserted in the neighbor table, the link estimator module will start calculating the 1-Hop ETX value to that neighbor as described earlier.

3.3.1 The Neighbor Table and the Routing Table

The routing table differs from the neighbor table in terms of the ETX value. The entries in the routing table the multi-hop ETX values to the sink; the multi-hop ETX value, from the current node towards the sink, is included in its routing beacons. Once a new neighbor is inserted in the neighbor table, and its 1-Hop ETX towards that neighbor node is calculated; the node will calculate the overall cost towards the sink (multi-hop ETX) and insert it in the routing table. It is clear that both tables are closely related.

Adding or removing a neighbor from the neighbor table follows certain conditions. If there is still space in the neighbor table, and the node receives a beacon from a new neighbor it will just add the new neighbor to the neighbor table. If the table is full, and the new neighbor node has to be inserted, it has to be checked if one of the entries in the table can be removed and replaced by the new neighbor [37].

There are only two cases in which neighbor nodes can't be evicted: if the neighbor node is pinned with an ETX value of 0 (sink node), or the neighbor nodes is pinned as a parent node.

The other entries that are not pinned can be removed under the following conditions [37]:

- First condition of removing one of the neighbor nodes from the table is not having any update for a fixed timeout; this flags the node as invalid and ready for removal if a new neighbor becomes available. This feature is important in the case of mobile networks. This timeout can be changed in order to keep up with the dynamic topology of the network.
- Another condition for the eviction of one of the neighbors is not having a value for Q_u or Q_b yet, which flags the node as not mature and hence possible to evict.
- If all the neighbors in the neighbor table are valid, mature and not pinned, but there exists a neighbor with a 1-Hop ETX value that is higher than a pre-defined threshold value then it can be evicted.
- If there is no neighbor to evict, there are two cases where an eviction is forced: if the incoming beacon is from the sink which is not yet in the neighbor table or if the overall path ETX of the new neighbor is lower than at least one of the current nodes in the neighbor table.

Once the node has neighbors in its neighbor tables, and the 1-Hop ETX is calculated, the overall ETX from the current node can be calculated and used in the routing table to start choosing a parent.

The parent is chosen based on the lowest overall ETX value to the sink. Once a parent is chosen it will be pinned and can't be removed from the routing table unless it becomes unpinned again. The parent update procedure happens either periodically every 8 seconds, as in the TinyOS implementation and the Castalia Implementation, or it can be updated in one of the following events [37]:

- The node sends a beacon. In case of the trickle algorithm when the node send out beacons closely spaced the parent will be evaluated and updated every time a beacon is sent out.
- The parent becomes invalid due to loss of updates. This happens when there are no beacons or acknowledgements received by that parent in a specified timeout.
- The current parent becomes congested; it sends a beacon with the congestion flag set to 1, the neighbor nodes will change to another parent even if the current parent has the lowest ETX value to the sink.
- If one of the neighbors is not congested anymore: this means that the neighbors were not choosing that neighbor as a parent, and after the congestion is cleared they can use it as a parent.
- The node has no path to the sink, this mainly happens when the node is newly attached to the network, the network is in its set up phase or the node was disconnected from the rest of the network and didn't receive any beacons from the neighbor nodes. It will continuously check its routing table for a parent until one is available.

3.4 Advantages of CTP

There are several reasons why CTP is of advantage for the collection of information from the sensor nodes in the environment, and why it is considered efficient in static wireless sensor networks:

Network Expansion

Sensor networks consist of a number of nodes put together to gather information from the surrounding. Sometimes, it is required to add more nodes to the network and CTP makes it easy to add nodes without the need for a manual change in all other nodes. The new node will receive beacons and just attach itself to the tree.

Loop Management

It happens that nodes choose a wrong parent and the packets get forwarded in a loop. In CTP, because both beacons and data packets include their multi-hop ETX value to the sink, the receiving node can compare the ETX value of the incoming packet to the values in its routing table. The node will discover a loop in the network and request the neighbor nodes to reset the beacon interval.

Trickle Algorithm

Control overhead reduction is important. It has to be considered in networks with limited recourses. CTP controls the frequency in which beacons are sent; the interval between the different beacons increases from a minimum interval until it reaches a maximum interval. The beacon interval gets reset to the minimum in the following cases [37]:

- If no path to the sink exists. The node will send out its packets with a path request. All neighbor nodes overhearing the path request will reset the beacon interval.
- If a neighbor parent node is congested it will send out packets with a congested pin. The neighbor nodes will search for a new parent to reduce traffic on the congested parent. Each of the neighbor nodes that overhear the congested flag will reset their beacon interval.

- If a loop is detected by receiving a packet from one of the node's child nodes with a lower ETX indicating a loop, the nodes will trigger a route update and the neighbor nodes will reset its beacon interval.

Tree Maintenance

If a node dies, or gets disconnected the tree doesn't disconnect completely. Only the disconnected node is affected; other links and the existing tree remain. There is no full tree recreation; each node individually requests a route if it can't reach its parent node. Every node that hears the route request will reset its trickle algorithm but the neighbor entries in the routing table will not be reset.

3.5 Fixed Node Assisted - Collection Tree Protocol

Node mobility in wireless sensor networks can cause a significant decrease in the network's performance, especially if the protocol is not configured to quickly handle link and path breakages. CTP has mechanisms in place to handle loops and lost parents. However, even though CTP performs better than AODV, the PRR dropped when CTP was tested in a mobile network [8]. In order to better handle mobility FNA-CTP was proposed [8]; where a number of static nodes is introduced in the network that act as backup nodes in case of missed retransmissions to mobile parent nodes.

3.5.1 Difference to Standard CTP

There are some differences between CTP and FNA-CTP, those differences Sharma et al [8] explain in detail. The following will outline the differences:

Parameter tuning

A number of parameters have been adjusted in standard CTP to increase the performance and lower the overhead in the network:

- The buffer size of the fixed nodes, in order to handle more packets that are forwarded from mobile nodes.
- Number of retransmissions for the mobile node has been decreased, because it was a significant cause for overhead.
- The mobile nodes send the beacons in fixed intervals reduce control overhead. The nodes are still able to retrieve route information from data packets.

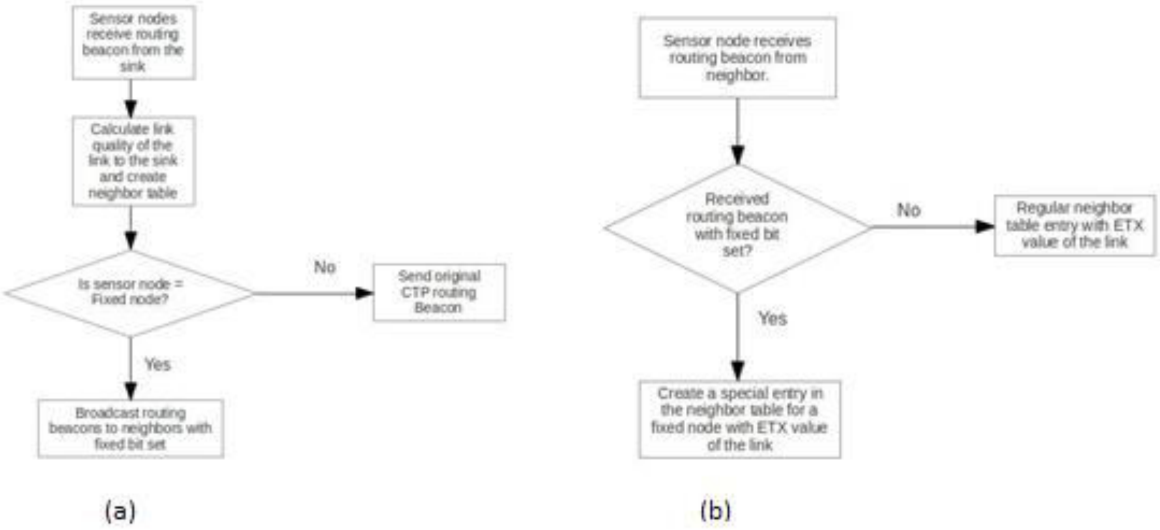
Routing Beacons

When a node is fixed, its outgoing beacons will include a new flag that identifies the node as a fixed node. In standard CTP, the beacons had unused bits in the header intended for future use; the standard protocol only uses the pull flag and the congested flag. When other nodes hear the beacon from a fixed node, they will add it as a special entry in the routing table. However, this doesn't mean that the fixed nodes are forced to be added into the routing table; FNA-CTP doesn't change the conditions for the eviction of the routing table entries as specified per standard CTP. FNA-CTP as well doesn't enforce the fixed node entry to be kept in the routing table; the only condition where a fixed node is pinned and hence can't be removed is when it is currently used as a parent node for the mobile node. After reaching the maximum number of retransmissions towards a mobile parent node, the node will look in its routing table for a fixed node, if available it will switch the parent to the fixed node and pin it. From here, standard CTP procedure will apply.

Retransmissions

If the current parent node is a mobile node, the node will try to forward packets to it. If the packet transmission is unsuccessful, the node will attempt to retransmit the nodes a number of times until it reaches the maximum number of retransmissions set by CTP. If the maximum number of retransmissions is reached, the node will not drop the packet as per standard protocol. Instead, the node will search for a fixed node in its routing table, if found it will attempt again to retransmit the packet to the fixed parent, but if that fails then the packet will be dropped as per standard protocol.

The following flow diagram shows FNA-CTP.



(a) (b)

Figure 2.1: Sending and receiving of a fixed node beacon.

3.5.2 Difference to Clustering Based Routing Protocols

The static nodes introduced in the network are placed such that they cover majority of the network. One can argue about the difference to a clustering based routing protocol. FNA-CTP doesn't force the nodes to use any node as a parent; the nodes can choose a parent node based on the link quality to it. In fact, the fixed nodes can use mobile nodes as parent nodes if they have better paths to the sink. While in clustering protocols the nodes have to send their packets to the cluster head without looking at the link quality.

Except for the static node flag, and the retransmission of the packets to an available static node, CTP's execution in each node remains the same.

Chapter Four
Network Model

This chapter briefly describes Castalia, a wireless sensor network simulator. The network model that has been used to simulate CTP and FNA-CTP in this research is explained. Each layer of the model and all the parameters required to build the full network are described. Further on, the main performance metrics that were used to compare and analyze the performance of the networks are listed and defined. Finally, the parameters for all different simulation scenarios are provided in form of tables.

4.1 Castalia

In this study we used Castalia, an Omnet++ -based wireless sensor network simulator. It can be used to test algorithms in realistic wireless channel and radio models. The inventors of Castalia based their data on empirical results from real time experiments. Castalia has a number of routing and MAC layer protocols implemented such as CTP. Moreover, it makes it easy for the user to extend and modify existing protocols. As well as the ability to implement different mobility models.

The channel model is a complex model that takes into account temporal variations of the path loss. It has as well several options for calculating interference between the sensor nodes [29]. The original simulator supported only a linear mobility model. However, we extended it by adding Random Waypoint mobility model for the purpose of this research which will be discussed in more detail in section 4.4.

The radio model is based on real radio specifications of the cc2420 chip. Different levels of transmission power as per standard were implemented as well. The different state matrix for the node, such as transmitting, receiving and idle power consumption is available to the user, as well as the power consumption matrix for the transition between each of these states. The simulator as well has different MAC protocols available based on various standards such as CSMA with and without duty cycle, and 802.15.4.

Finally, the main programming language used in the simulator is C++. It makes it easy to the user to modify an existing protocol or to implement a new protocol; whether it is at the routing layer, the network layer, a different application protocol or a mobility model.

4.2 CTP for Castalia

CTP for Castalia mimics the implementation of CTP in TinyOS. However, there are some differences to the TinyOS version.

In Castalia's CTP implementation, a modified MAC layer is modeled that mimics the TinyOS MAC layer. Castalia's original MAC layer does not provide link layer acknowledgments or packet spoofing to listen to route requests from other nodes. However, the available version of CTP for Castalia doesn't support the wake-up, sleep or idle states of the node. The node is switching between transmissions or receiving only. However, this issue is not a factor in our model as we are not conducting full power consumption and network life time analysis that might be affected by the cycle of the sensor node.

4.3 Default CTP Parameters

Since FNA-CTP is a modified and enhanced version of CTP, there are certain parameters of CTP that remain the same as per standard. In the following sections, we will outline the important default CTP parameters that we used in our network model.

4.3.1 Radio Model

Each sensor node is assumed to have an omnidirectional antenna in all simulation scenarios. The radiation pattern used is based on the log shadowing model that is based on the following equation [37]:

$$PL(d) = PL(d_0) + \mu. 10 \log\left(\frac{d}{d_0}\right) + X_\sigma \qquad Eq.6$$

Where

- $PL(d)$ is the path loss at distance d.
- d_0 is a reference distance usually assumed to be a unity disk of value 1.
- $PL(d_0)$ is the path loss at d = 1 which is equal to 54.2247 dbm.
- μ is the path loss exponent and is equal to 2.4 in our model.
- X_σ is a Gaussian zero mean random variable that reflects the attenuation caused by flat fading, and because to fading is assumed in this model this variable is negligible in our model.

By using equation Eq.6, we can calculate the transmission range of a sensor if we know the sensitivity of the receiving nodes and the transmitter power Tx of the sending node.

For example, if the Tx power of the sending node is -5 dbm and the receiver's sensitivity is -95 dbm, then the transmission distance of the sender node will be calculated using the previous equation as follows:

-5 – (-95) = 54.2247 + (2.4) * 10 Log (d)

Then d would be approximately 31 meters.

The interference in our model is an additive model, each neighbor node's radio signal is counted as an additive noise to the thermal noise that might interfere with the successful transmission of a packet.

4.3.2 MAC layer

The main parameters of the MAC layer are the back off timers, the initial and the congestion back off timers. The first timer is for the first attempt to send a packet, while the second timer is used when the channel is sensed to be busy and hence a back off is required.

Important as well is the time-out after an acknowledgment, before requesting to retransmit the packet from the routing layer. These values play a role in the overall end to end latency of receiving a packet.

The values for the MAC layer [37] are in Table 4.1.

Table 4.1: Parameters of the MAC Layer

Parameter	**Value/Unit**
Initial backoff window	0.3-10 ms
Congestion backoff window	0.3-2.4 ms
Acknowledgement timeout	7.8 ms

4.3.3 The Routing layer

CTP is implemented in different modules, and each of these modules has his own parameters. The following sections will describe each of these components separately.

4.3.3.1 Link Estimator (LE)

As mentioned earlier, the LE is responsible for calculating the 1-Hop ETX value. Table 4.2 shows the default values of CTP that are used in our model. The table shows the parameters that have been used in eq. 5, as well as the window sizes for calculating the outgoing and incoming link qualities. The default size of routing table is as well defined in the LE module [37].

Table 4.2: the parameters of the LE

Parameter	Value/Unit
α_{ETX}	0.9
w_b	3 packets
w_u	5 packets
Size of the table	10 entries

4.3.3.2 The Routing Engine (RE)

The default parameters of the routing engine [37] used in our network model are shown in table 4.3

Table 4.3: Parameters of the RE

Parameter	Value/Unit
Size of routing table	10 entries
Parent periodical refresh period	8 s
Minimum length of beacon interval	64 ms
Maximum length of beacon interval	250 s

The RE is as well responsible for controlling the beacon interval, which has been changed in our model from the default value and will be shown prior to each simulation scenario. It is responsible as well for the periodical refresh period, to search for a better parent in the routing table.

4.3.3.3 The Forwarding Engine (FE)

The default CTP parameters for the FE are outlined in table 4.4. The FE is responsible for retransmitting the packets until it reaches the maximum number of retransmissions [37].

Table 4.4: default parameters of the FE

Parameter	Unit/Value
Forwarding queue size	12 packets
Sent cache size	4 Packets
Maximum Number of Retransmissions	30

4.4 Mobility Model

There are several mobility models available to be used to simulate a mobile network. Our model is a general model, and hence not application specific, we make use of the Random Waypoint mobility model [32]. Each node moves in a randomly chosen direction for a specified move time, then stops for a specified stop time. The node may change its direction after the stop time or continue in the previous direction. If a node reaches the border of the network it will reflect and change its direction accordingly.

Random waypoint mobility model can be considered a worst case scenario. It is the most common mobility model used in simulation scenarios; the nodes can move freely in the network. The user doesn't know how the nodes move. When repeating the simulations the movement pattern will not be the same in each run as well as the starting node positions. This mobility model makes it difficult to fully design a network, such as choosing an appropriate sink position to be able to collect the maximum number of application packets from the sensor nodes. The parameters of our model are shown in table 4.5.

Table 4.5: Parameters of the mobility model

Parameter	Unit/Value
Node Speed	1,3,5 meters/second
Move time	30 seconds
Stop time	10 seconds

4.5 Application Layer

In this model, we assume that the sensor node is making a snapshot of the physical environment every t second. Which is then sent to the routing layer to forward it to the sink.

4.6 Performance Metrics

In order to be able to investigate the performance of a protocol, certain performance metrics are used to investigate the effect of the protocol changes on the network. These metrics will provide an insight on the usefulness of the suggested changes; in this case, they are the deciding factor in the suggested guidelines for an FNA-CTP network design.

4.6.1 Packet Reception Ratio PRR

The PRR is a measure for the successful transmissions of the data packets in the network. If N_{total} is the total number of packets sent by all sensor nodes during the simulation, and $N_{received}$ is the number of data packets received at the sink not including duplicate packets.

Then the PRR is calculated as follows

$$PRR\,\% = \frac{N_{received}}{N_{total}} \times 100 \qquad Eq.7$$

The PRR is the main metric used in deciding if the network is performing well or not. CTP promises a nearly 100% PRR in static scenarios [37].

4.6.2 Overhead

Overhead can be caused by many factors, in our case it could be the control overhead due to control beacons, retransmission cost overheard due to the retransmission of unacknowledged packets. Duplicate packets circulating in the network due to retransmissions will cause overhead as well.

4.6.2.1 Control Overhead

This is the number of beacons sent by all the nodes in the network to build and maintain the collection tree. Control overhead is associated with the cost and energy required from the network to achieve a certain PRR.

4.6.2.2 Retransmission Cost Overhead

Due to the frequent link breakages in the mobile network, there are a number of lost packets/acknowledgements which force the nodes to retransmit the packets in order to successfully forward it. CTP's default maximum number of retransmission is set to 30, which will cost the node high energy and congestion in the network if each packet requires a high number of retransmissions.

4.6.2.3 Number of duplicate packets

When a packet is retransmitted many times due to lost acknowledgement or lost packets, there is a number of duplicate packets in the network. CTP's FE is responsible of suppressing duplicate packets by checking a sequence number of the packets that is given to it when it is transmitted.

However, due to the frequent parent changes in a mobile network, it can happen that the same packet is sent out to two different parents. For example, when the first parent is not responding, and during the retransmissions, the node chooses a different parent to forward it's packet to it; it is then still possible that both parents receive the same copy of the packet and then forward it to the sink.

4.6.3 Average and Maximum Hop Count

In order to evaluate the protocol, the average and the maximum number of hops it takes for a packet to reach the sink becomes important. It is as indication of how long the packet had to transfer in the network until it reached the sink, which is related as well to the delay of the packets.

4.7 Other Definitions

In this research, in addition to the number of nodes and the network size, there are a number of other parameters that were used. Here we will define these used parameters:

4.7.1 Node density

The node density of the network here is defined as the number of nodes/m^2, not taking into consideration the transmission power of the nodes. The network can be sparse with only a few nodes or very dense with a high number of nodes.

4.7.2 Fixed Node Ratio

The Fixed Node Ratio FNR is a measure of what percentage of nodes in a network is static (non-mobile). FNR is represented as (The number of Fixed Nodes to the total number of Nodes). This value can be represented as a percentage as well, but we chose this format in order to quickly understand the approximate positions of the fixed nodes which are placed on a grid.

Chapter Five

CTP's Performance in Mixed Mobile-Static Networks

In this chapter, we investigate the effect of introducing a set of fixed nodes to CTP, without changing any of its default parameters or operational procedures. The performance of a mobile network, running CTP and having a set of fixed nodes, was thoroughly studied. Sections 5.1-5.3 provide simulation results and their analysis for a mixed mobile-static network running standard CTP. Section 5.4 compares the results of having 9 fixed nodes in the network when running standard CTP and FNA-CTP.

5.1 CTP with Fixed Nodes

It has been shown [38] that CTP performs well in a fully static network; the network's PRR can reach up to 98% without high control overhead due to the trickle algorithm. Hence, it is only logical that having a mix of static-mobile nodes will show an increase in the performance over an all-mobile network. However, this doesn't mean that placing a number of fixed nodes randomly in the field, such as at the edges of the field, will increase the performance. Sharma in his work [39] proposed that the static node's transmission range should cover most of the field.

Table 5.1: Network parameters of the initial CTP network with added fixed nodes

Parameters	**Unit/Value**
Field parameters	
Network size	100 by 100 meters^2
Number of nodes	40 nodes
Sink position	(50,100) Top center of the field
Application layer parameters	
Data traffic	0.333 packets/second
Fixed nodes	
Number of fixed nodes	6,7,8,9 nodes

We started by a network of 100 by 100 meters^2 with 40 nodes. The parameters are shown in table 5.1. Then, we tried four different scenarios with a different number of fixed nodes placed on a grid for simplicity. The used configurations are shown in figure 5.1; each configuration was named CTP-(Number of fixed nodes) to distinguish between them.

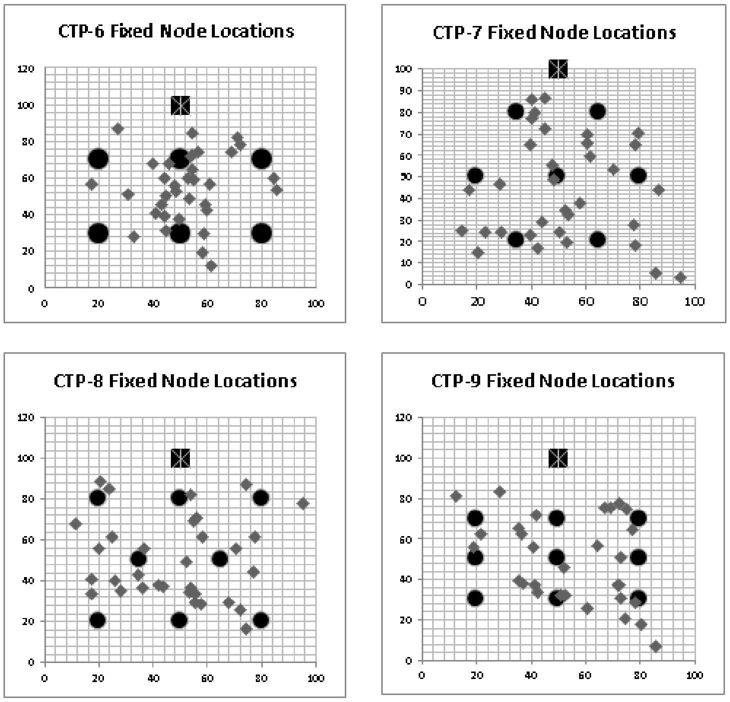


Figure 5.1: physical location of the fixed nodes in the different configurations (in meters).

5.1 Performance Results

Figure 5.2 shows an increase in PRR, compared to the all mobile scenario, when there are a number fixed nodes in the network. The simulation results showed that the configuration of CTP-9 had the highest PRR with an improvement of 40% over the all mobile scenario. Moreover, figure 5.2 shows a variation in PRR between the different configurations, this is due to the node locations in the network.

As we will outline later, the distance between the fixed nodes and the sink, as well as their distance to each other have an effect on PRR. For example, CTP-7 and CTP-8 had lower PRR, which is due to the distance of the fixed nodes to each other and to the sink; both scenarios had a higher number of packets dropped with interference and thermal noise than the rest of the scenarios. Moreover, there were a higher number of packets dropped due to a busy client. This was the reason to the lower number of received packets.

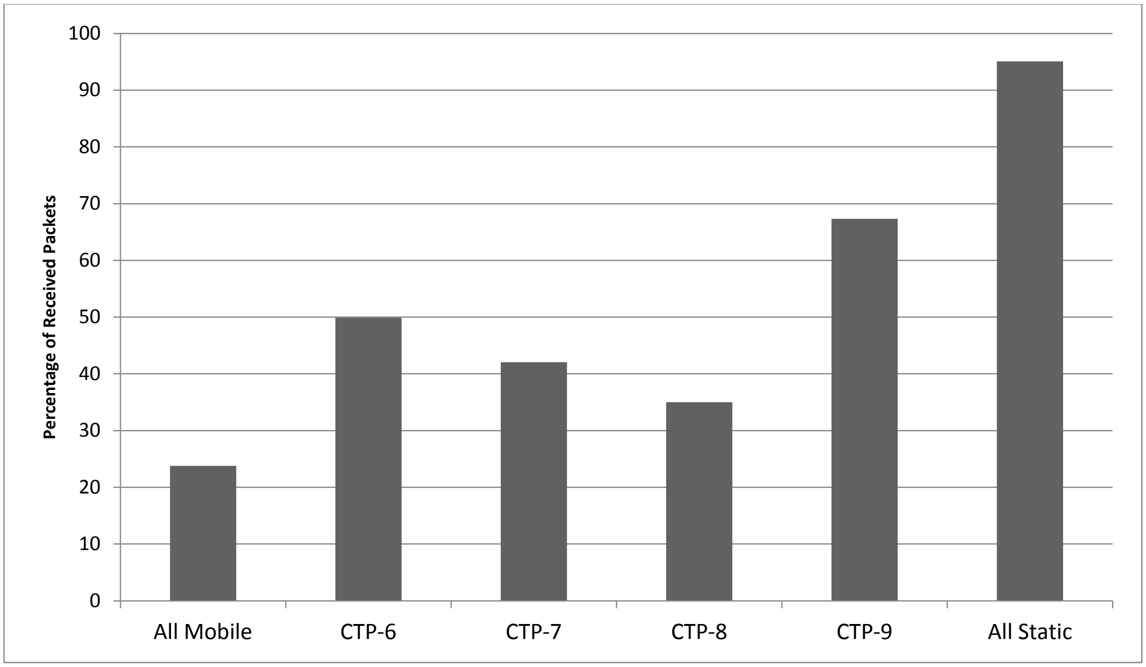


Figure 5.2: Packet Reception Ratio for different combinations of static-mobile networks.

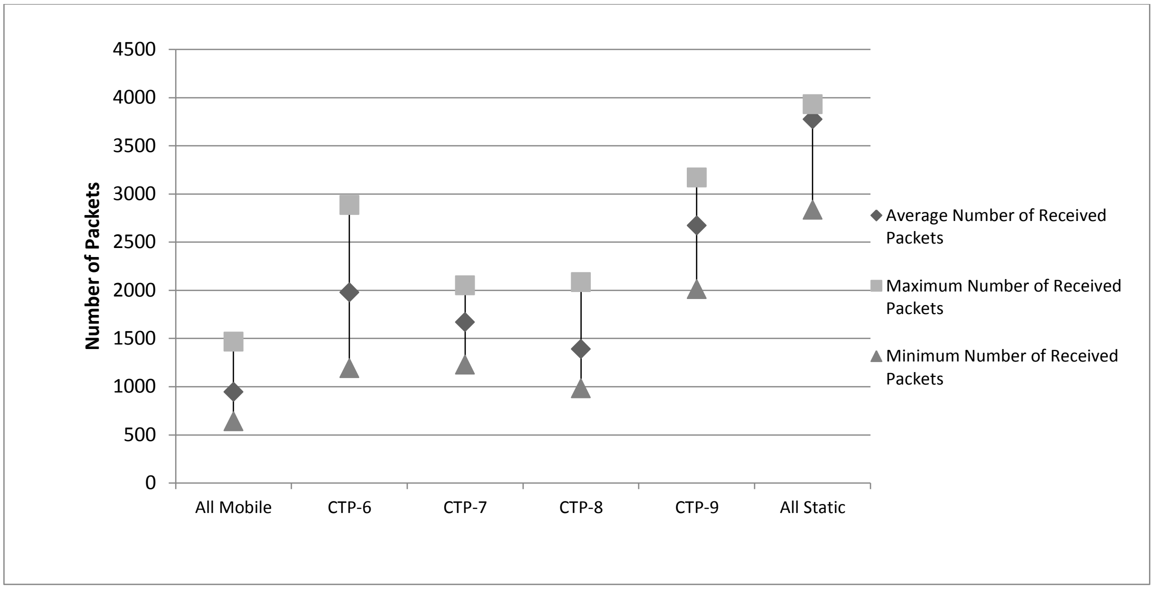


Figure 5.3: Average, maximum and minimum number of received packets.

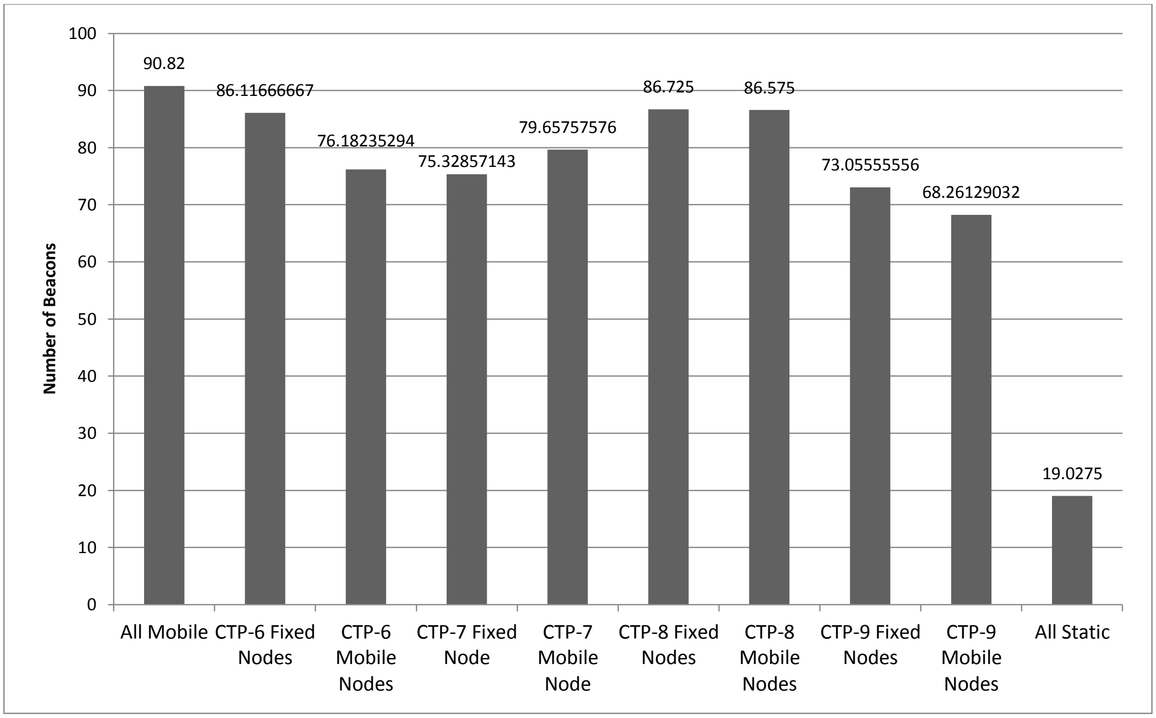


Figure 5.4: Average number of transmitted beacons per fixed and mobile node.

Mobility in the network has a significant effect on the frequency of beacons used to maintain the tree. The instability of the network causes congestions and loops, these changes cause more path requests. The pull request has the effect of resetting the beacon interval to the minimum, hence more control overhead. Figure 5.4 shows the lower number of control beacons in the fully static network; it shows how the trickle algorithm is reducing the overhead. The number of beacons, in all other scenarios, is at least 3 times higher. Node mobility minimizes the benefits of the trickle algorithm in reducing the volume of routing beacons, because the network is not stable. However, it shows as well that the protocol is reacting to the link breakages and resetting the trickle interval to update the routing table. Moreover, CTP continuously tries to find the path with the best ETX. However, due to the frequent beacons and ETX recalculations there will be frequent parent changes. The network with configuration CTP-9 had a lower number of parent changes when compared to the other configurations as shown in figure 5.5.

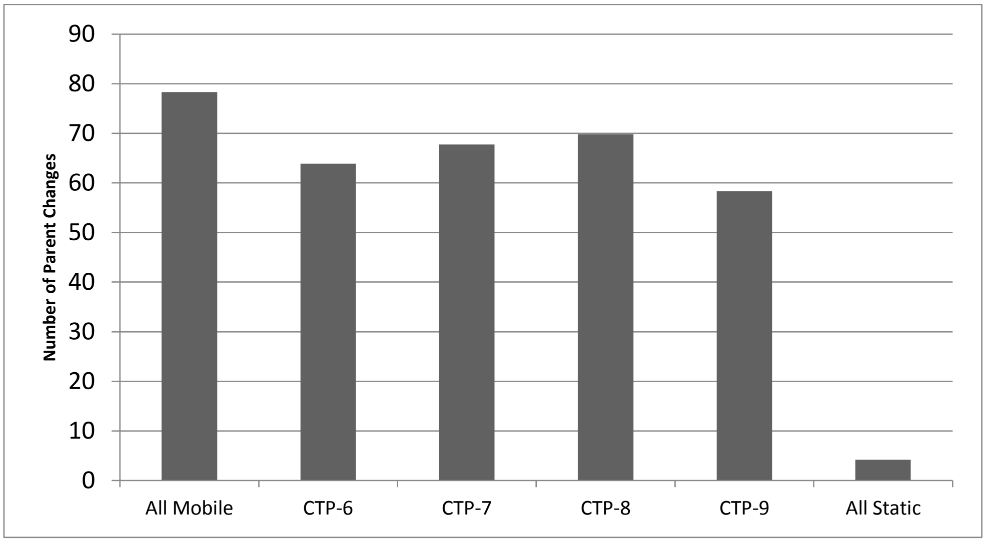


Figure 5.5: Average number of parent changes.

Another source of overhead in the CTP network is the retransmission when not receiving acknowledgments. The “All Mobile” scenario suffered from a high number of dropped packets because the maximum number of packet retransmissions limit was reached. The simulation results show that the number of packet retransmissions per node is in fact more dominant than the control overhead. In the “All Mobile” scenario, about 2100 packets were retransmitted on average per node in order to successfully send 100 packets on average per node. There are a high number of attempts to resend a packet, the buffer fills up with other unsent packets; the network is overwhelmed with retransmitted packets and acknowledgements. Packets that are not acknowledged will eventually be dropped, resulting in loss of data packets.

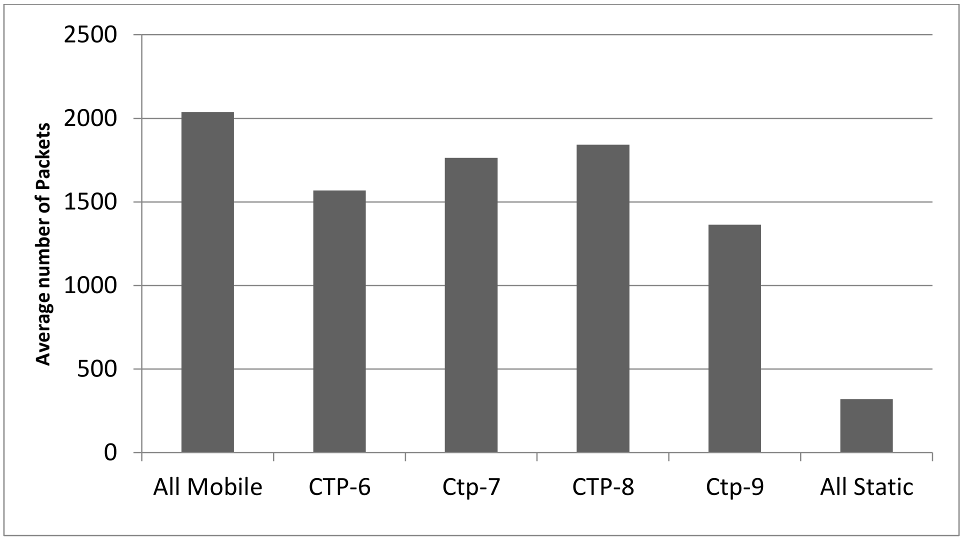


Figure 5.6: Average number of packet retransmissions per node for not acknowledged packets.

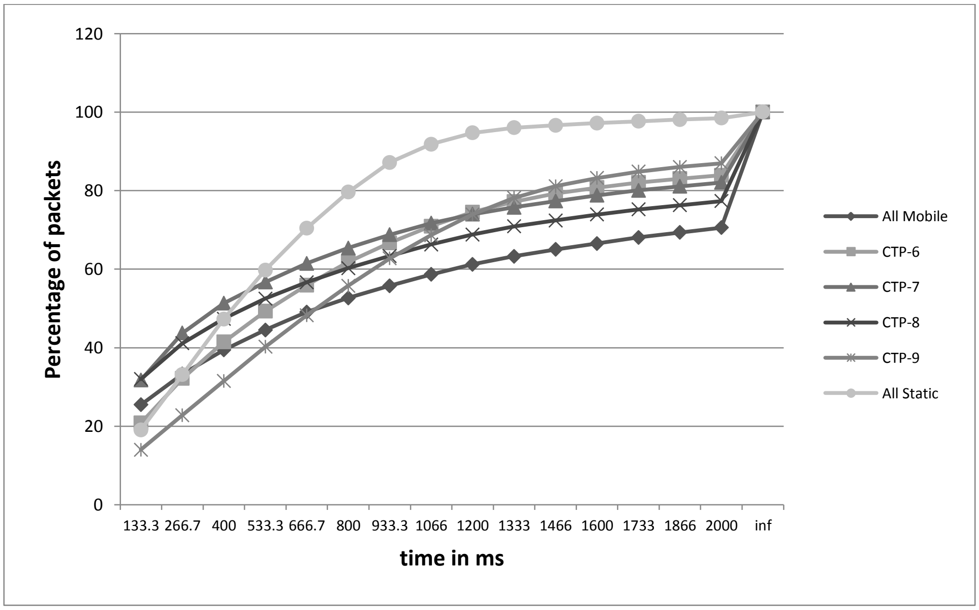


Figure 5.7: Cumulative distribution function of packets latency.

Packet latency is important in those scenarios when a quick response is required, and because there are several hops between the sensor nodes and the sink, latency becomes an issue in certain applications. In the static scenario 80% of the packets reached the sink in less than 800ms. There is a higher delay for all other scenarios; this is due to the high number of retransmissions. Moreover, frequent parent changes imply that packets are forwarded through more nodes, and hence more hops to reach the sink resulting in longer delays.

5.2 Results Analysis

The simulation results show that there is an increase in the PRR when some of the nodes in the network are static. This is an improvement of over 30% in some cases; it means that having fixed nodes in the network introduces some stability to the network. The fixed nodes act as connecting branches to the sink, especially when they are well interconnected. However, there is still an overhead in terms of packet retransmissions and routing beacons. The maximum number of retransmissions and the beacon frequency have to be adjusted in order to lower the overhead in the network.

Moreover, some of the fixed nodes had an increase in the number of dropped packets due to buffer overflow, especially when those fixed nodes forwarded a higher number of packets than other nodes. The fixed node that is closest to the sink is used as a parent by other fixed nodes in range in addition to the mobile nodes in the area. The likelihood that a fixed node forwards more packets than a mobile node is high. Therefore, there is a need for larger buffers for the fixed nodes.

The two major factors for overhead are the data retransmissions and the transmitted beacons. The frequent link breakages cause the trickle algorithm to reset and send out beacons more frequently. In a mobile network, the nodes are more likely to reach the maximum number of retransmissions and then drop the packet. There is a low chance to successfully transmit a packet even after 30 retransmissions; this indicates that the maximum number of retransmissions is too high for the mobile nodes and the fixed nodes.

Based on these results the drawbacks were addressed by modifying CTP's parameters in addition to having a number of fixed nodes in the network.

5.3 FNA-CTP network analysis

Sharma et al [8] showed in their results that FNA-CTP performs well in mobile networks without extra overhead. This section is an elaboration and an addition to his results and their analysis, the work in [8] doesn't explain the reasons for packet drops in the network. From the scenarios of section 5.1 the FNA-9 configuration was chosen for further investigation because it had the highest PRR results.
The parameters used for the FNA-CTP network are as shown in table 5.2.

Table 5.2: the parameters of initial FNA-CTP network.

Parameters	**Unit/Value**
Field parameters	
Network size	100 by 100 meters^2
Number of nodes	40 nodes
Sink position	(50,100) Top center of the field
Application layer parameters	
Data traffic	0.333 packets/second
Fixed nodes	
Number of fixed nodes	9
Number of retransmissions for fixed nodes	20
Minimum to maximum beacon	64ms to 250s
Sent cache size for fixed nodes	8
FE queue size for fixed nodes	24
Mobile nodes	
Beacon interval for the mobile nodes	15 seconds
Number of retransmissions for mobile nodes	5

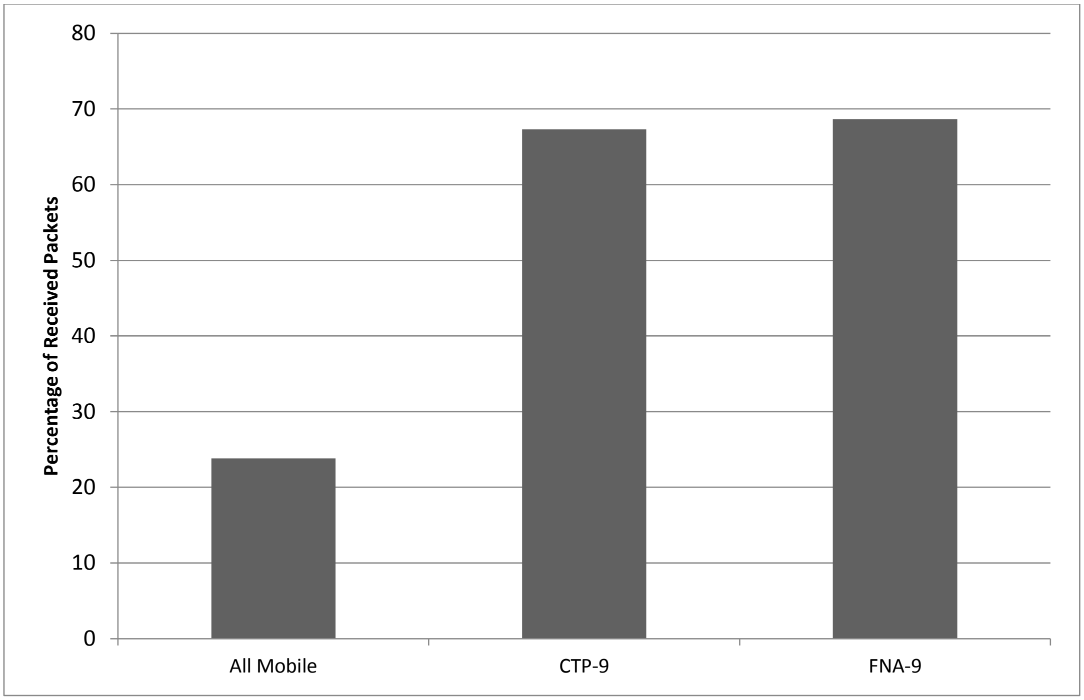


Figure 5.8: PRR comparison of three network scenarios.

The original CTP has a PRR of 23.4 % in the all mobile scenario, whereas for FNA-9 we observe a significant improvement of up to 68.7% in PRR. The PRR of the FNA-9 scenario is close to PRR of the CTP-9 scenario. It can be concluded, in these network scenarios, that the changes to the CTP algorithm required for FNA-CTP do not improve the PRR significantly over simply using fixed nodes with standard CTP.

Next, we compare the control overhead for the same scenarios. Figure 5.9 shows the average number of routing beacons transmitted per node. For the networks running standard CTP, the all mobile scenario had the highest number of transmitted beacons per node with an average of 90.2 beacons per node. For the FNA-CTP the average number of control beacons transmitted per mobile node was around 7. The average number of control beacons per fixed node was 18.6 on average. The total control overhead is significantly less in the case of FNA-CTP when turning off the trickle algorithm for the mobile nodes.

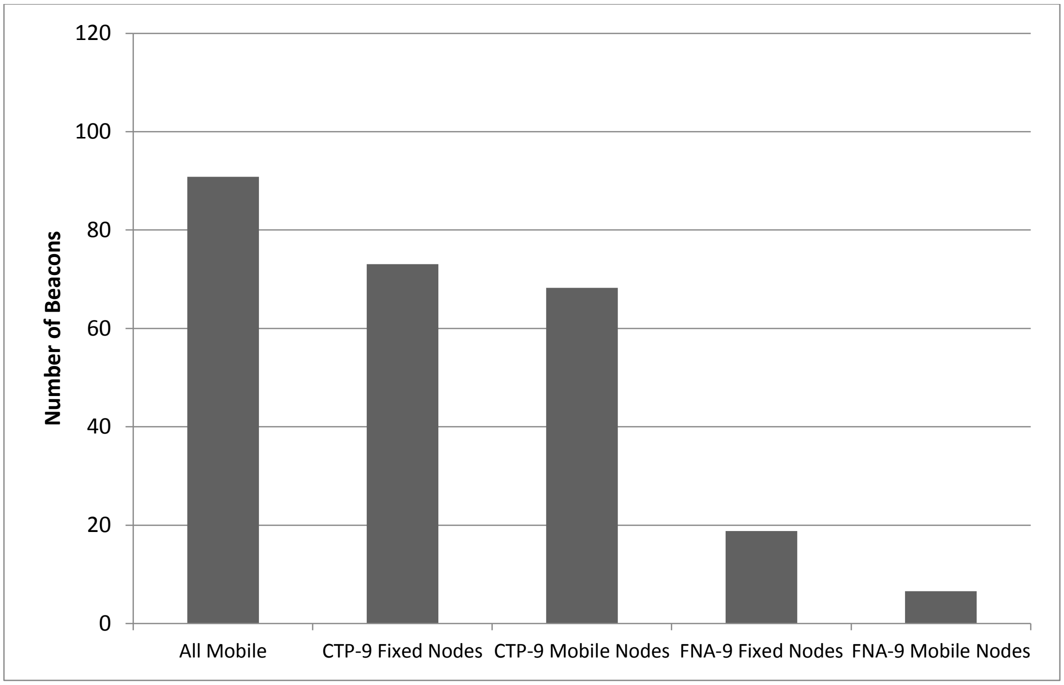


Figure 5.9: Average number of transmitted beacons per node.

Overhead due to packet retransmissions was very high in the all mobile scenario running standard CTP. The results in figure 5.10 show that the all mobile network had on average 2100 packets retransmitted, while the FNA-9 network had 500 packet retransmissions on average.

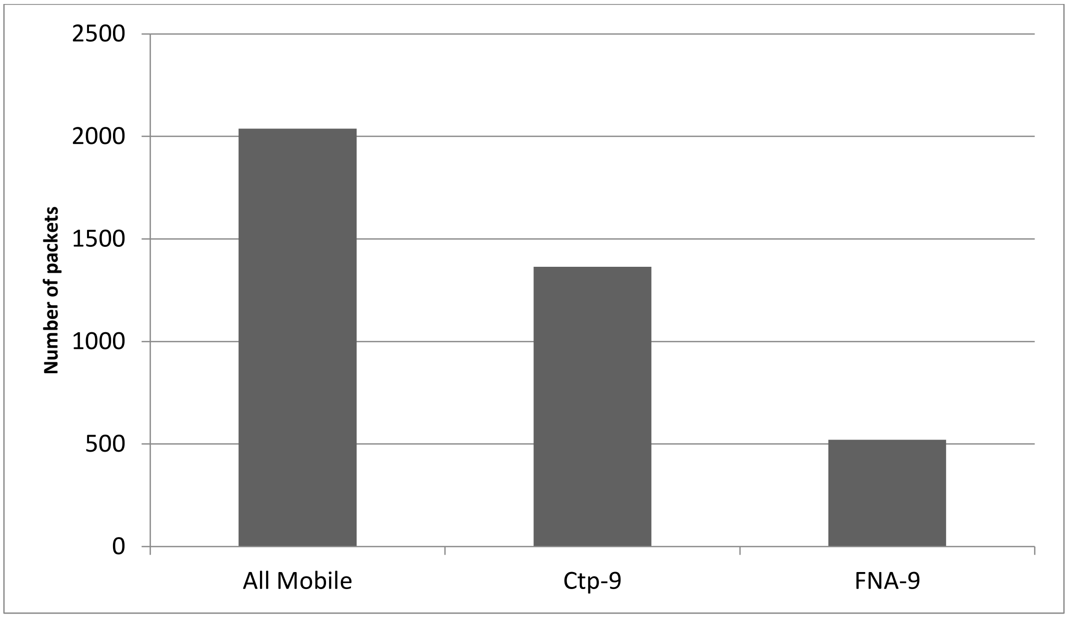


Figure 5.10: Average number of packet retransmissions due to lost acknowledgements.

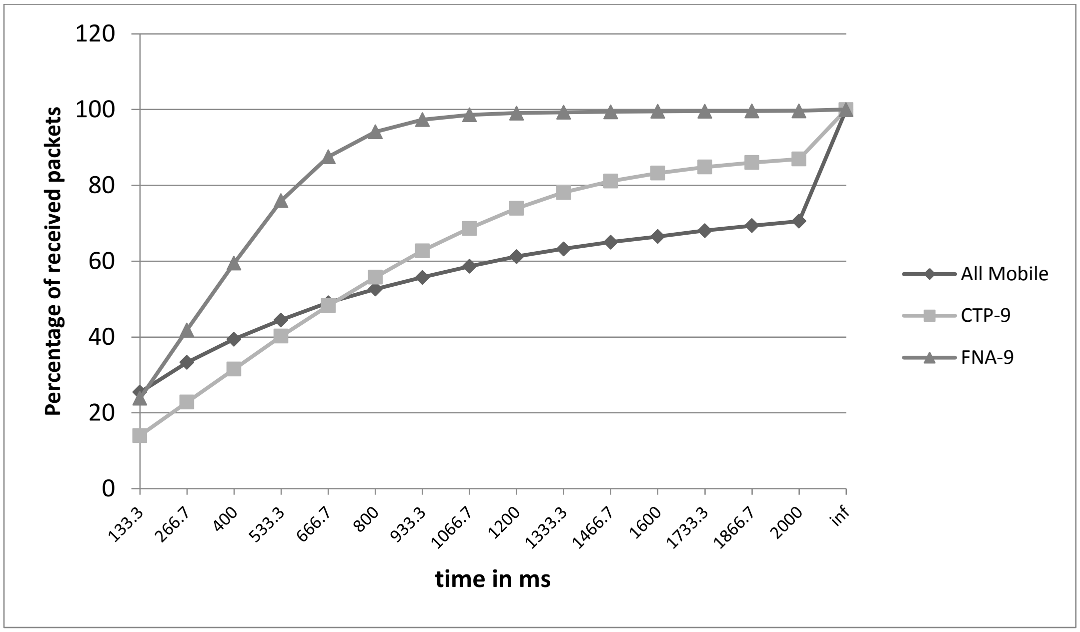


Figure 5.11: Cumulative distribution function of packets latency.

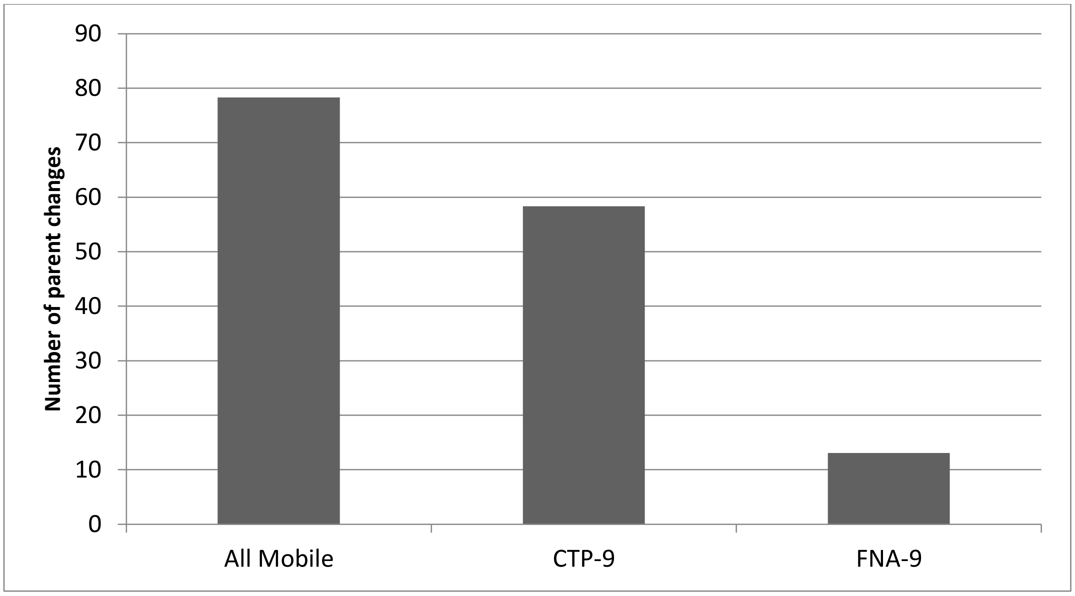


Figure 5.12: Average number of parent changes.

In terms of packet latency, figure 5.12 shows that by 800ms of simulation time, more than 80% of the packets were delivered at the sink.

Another change seen in the network is the number of parent changes, as shown in figure 5.12, the parent changes decreased from 78.7 to 13.4.

The number of packets dropped due to interference is reduced by more than 50% compared to the "All Mobile" scenario as shown in figure 5.13.

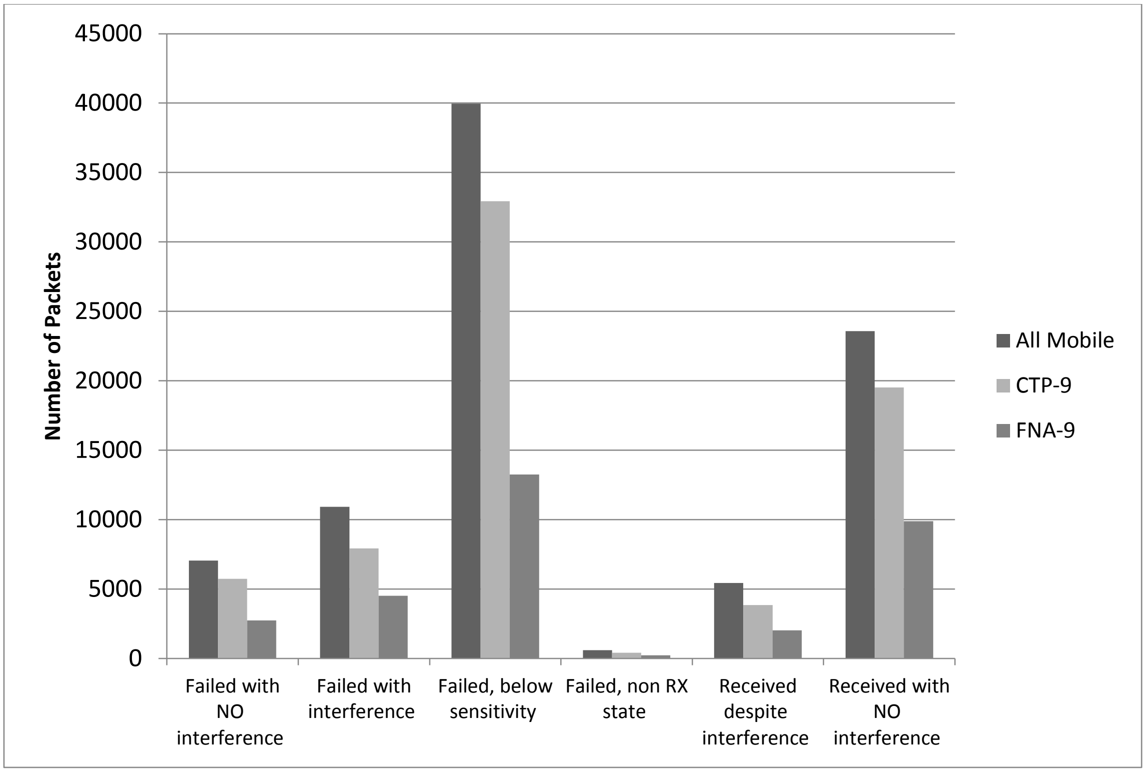


Figure 5.13: Breackdown of the received packets.

5.5 Results Analysis

The simulation results show that FNA-CTP outperforms CTP and CTP-Fixed in control overhead. Some packets were dropped because the buffer of the fixed nodes was filled up with unacknowledged packets, in particular when a mobile sensor node gets disconnected from its parent and its backup fixed node. This can be avoided by using fixed nodes with a higher transmission range.

The control beacons from the mobile nodes are programmed to be sent in fixed intervals, and every time a mobile node lost connectivity with its mobile parent it sent the packet to the fixed node. However, the static nodes are still sending beacons using trickle and hence can react to path requests.

However, the reliability achieved by adding a few fixed nodes at the cost of control overhead seems acceptable. In other words, the implementation of FNA-CTP in the network significantly stabilized the collection tree. Hence, it is important to vary the scenarios in terms of network size and number of nodes to investigate the use of FNA-CTP in larger networks. These scenarios will be investigated in more detail in chapter six.

Chapter Six

CTP and FNA-CTP Performance Sensitivity Analysis

This chapter discusses the results of different simulation scenarios running FNA-CTP. It extends the previous work in Ch.5 and by Sharma et al [8]; it studies the effect of varying the network size, the number of nodes, the node speed, the number of fixed nodes and the effect of their transmission power. The parameters for each scenario are explained at the beginning of each section. This chapter includes as well results analysis of all different scenarios outlining the factors that degrade or enhance the network's performance when having mobile nodes. Moreover, a look at the connectivity calculations and it's relation to PRR is provided in section 6.4.1, as well as the effect of varying the beacon interval in section 6.4.2.

6.1 All mobile CTP performance analysis

The first set of simulations, the all mobile scenario, was running standard CTP for three network sizes: small, medium and large size, with a different number of nodes per size of network. The goal is to see if the speed or the node density and the distance to the sink have an effect on the network's performance. The parameters are shown in Table 6.1.

Table 6.1: Parameters for the small sized network.

Parameters	Unit/Value
Field parameters	
Network size	50 by 50 m^2, 100 by 100 m^2, 150 by 150 m^2
Number of nodes	(3, 5,7,10, 12); (12,20,28,40,48); (27,45,63,90,108)

Sink position	(25,50);(50,100);(75,150)
Application layer parameters	
Data traffic	0.333 packets/second
Radio Layer	
Transmission power of nodes	-3dbm
Node Speed	1,3,5 m/s
Routing layer	
Beacon Interval	64ms to 250s

The following figures show the results of the all mobile scenarios.

6.1.1 Packet Reception Ratio

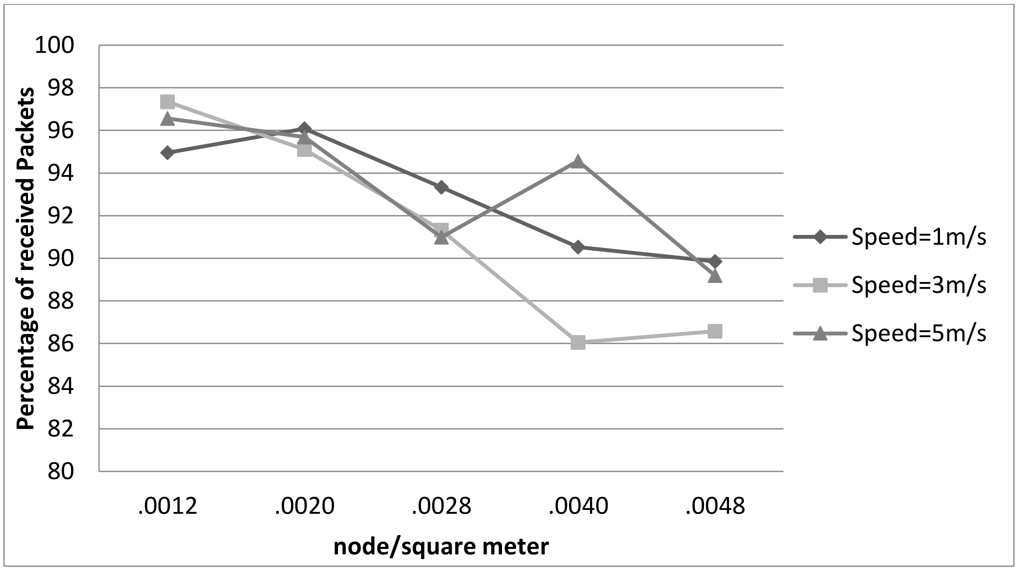


Figure 6.1: PRR of all mobile small size networks with different node speeds.

Figures 6.1-6.3 show the PRR at the sink for all different network sizes. For each network size, the node density and the node speed were changed. The results show that in a small sized network, where the nodes are considered close (mostly within one hop's reach) to the sink, the PRR is above 85% in all shown cases. Overall, in the small size network mobility doesn't affect the packet delivery to the sink drastically. However, the node density affects the PRR in the small sized network within a 7-11% range in this case.

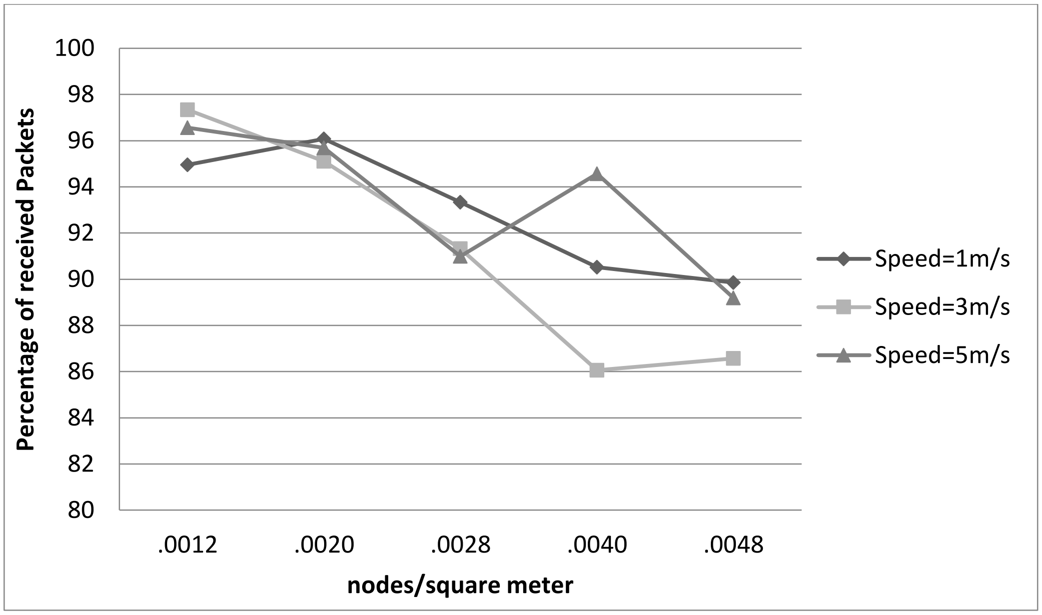


Figure 6.2: PRR of the all mobile medium sized network with different node speeds

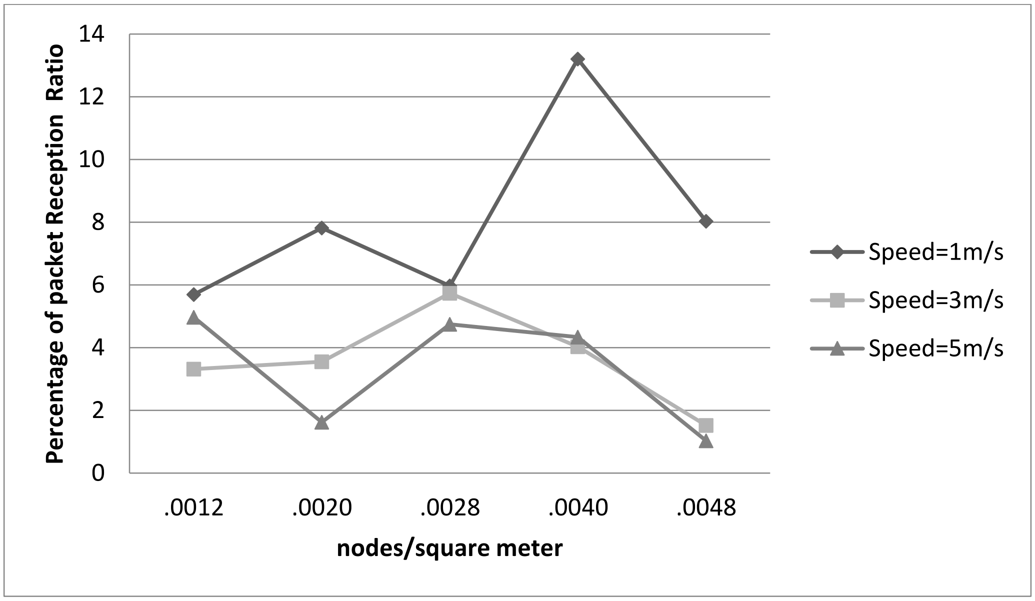


Figure 6.3: PRR of the all mobile large sized network with different node speeds.

In the medium sized network, where the nodes don’t always have a direct communication to the sink, both node density and node speed affect the PRR. There is a drop of up to 20% in PRR with the increase in node density. In the larger sized network, a significant decrease in PRR is observed. In this case, the distance between most nodes and the sink is more than 1 hop. The PRR in this case drops below an acceptable range (less than 20%). Node speed is another contributor to the network's performance; when the nodes move slowly with longer connection times, the network performs slightly better than when the nodes move with higher speeds.

The results show as well a spike in the PRR for the networks when the node density is 0.4% in the small network. This is due to the small network size and the nodes moving with higher speeds; the nodes have a higher probability to move into the sink's transmission range and have a direct connection to it.

In both the medium and large sized network, there is a performance increase for the same node density with the low speed. This means that the node density in this case and the low speed together perform the best in this all mobile scenario. Higher node density means more interference, while lower node density means lower connectivity and less alternate paths to the sink.

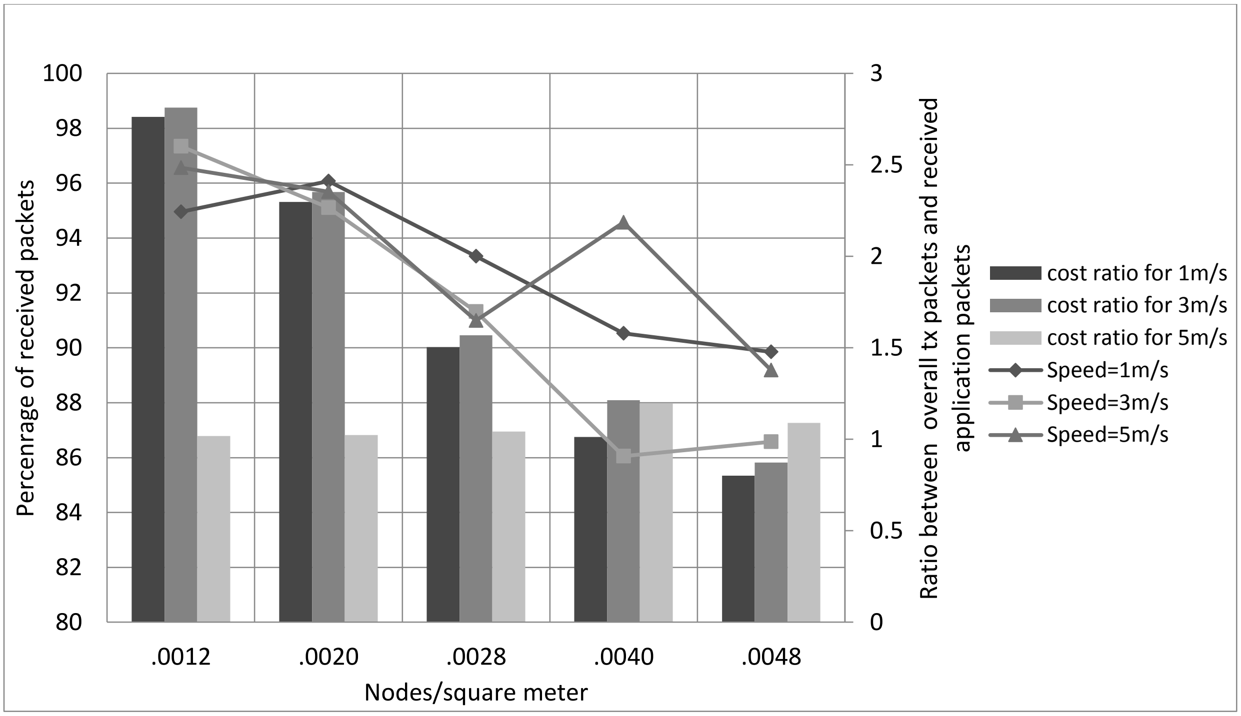


Figure 6.4: Comparison between PRR and cost ratio for the small sized network.

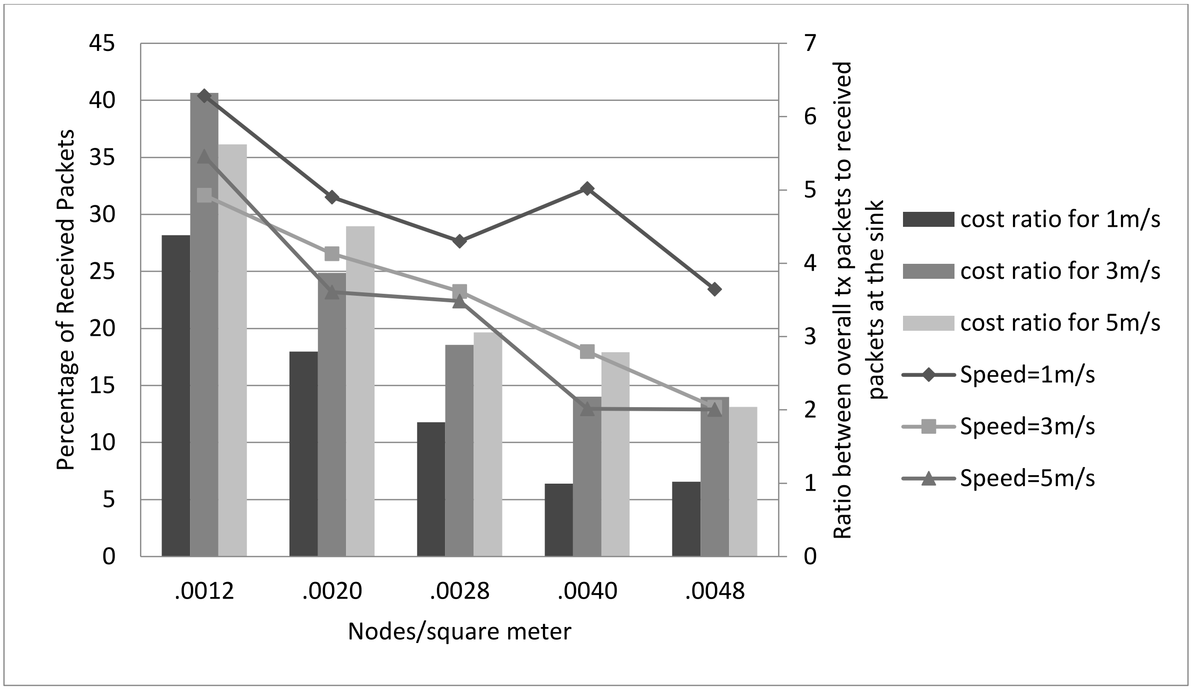


Figure 6.5: Comparison between PRR and cost ratio for the medium sized network

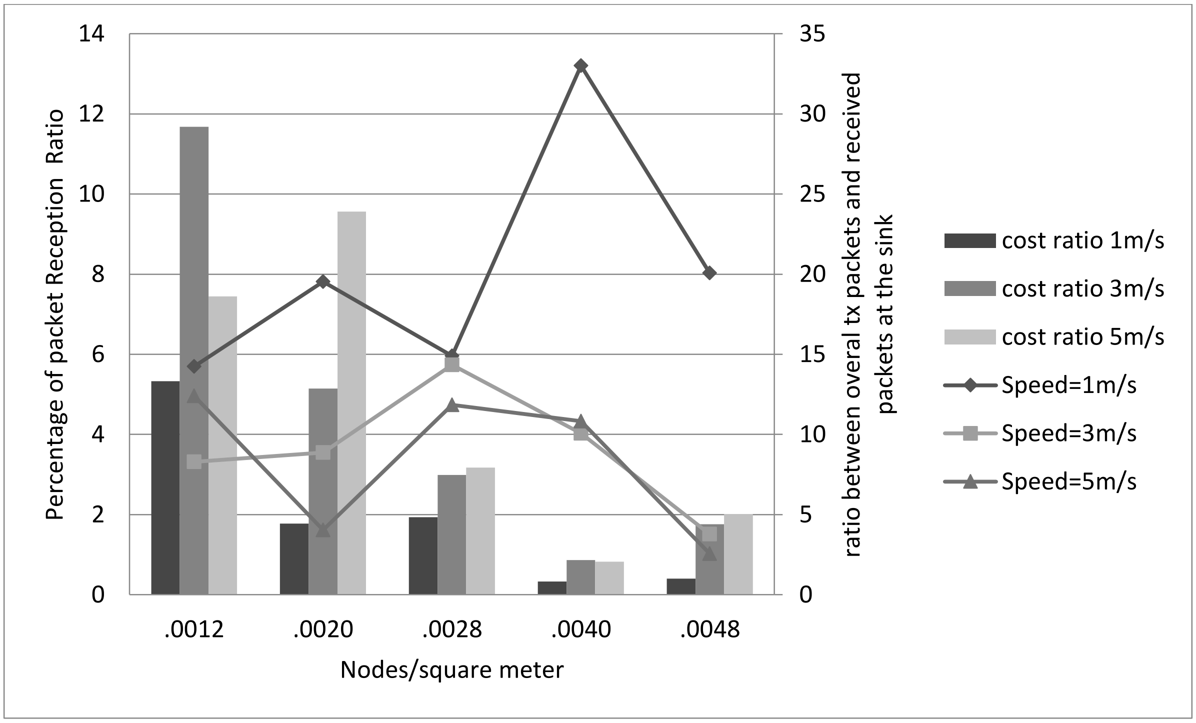


Figure 6.6: Comparison between PRR and cost ratio for the large size network.

Figures 6.4-6.6 show the cost, in terms of the number of overall transmitted packets in the network, in order to receive the resulting PRR in each case. It can be seen that the network with the lowest density had the best PRR. However, the results show that, in order to achieve the higher PRR, the nodes send up to 2.8 times the packets they actually have to send on average for the sparse network. With the increase in network size, the cost ratio increases with a decrease in the network's performance. Moreover, the results show that for a lower node density the network's overhead is higher.

The previous cost ratio takes into consideration any type of packets sent out by the node, this includes application packets and their retransmissions, acknowledgments and control packets. So, in order to better understand the low performance in the network, we will first study the different network layers and the main reasons for lost and dropped packets in each of them.

In the large sized network the PRR was very low. When looking at figure 6.6, low speed and a higher node density caused a spike in the PRR. The higher speed network with the density 0.0028 nodes/m^2 performed better than the rest of the scenarios. It had lower interference and less dropped packets than the rest of the scenarios as it will be outlined in section 6.2.1.

6.1.2 Packet Drops at the Radio Layer

First, a look at the results of the radio layer is taken, to check the packets dropped due to interference or without interference. The network with lower node density would be desirable from the cost point of view; in cases where a higher number of nodes exist, the effect of the interference between them becomes important.

Figure 6.7 shows the percentage of packets dropped without interference; castalia counts these packets as dropped due to thermal noise and Bit Error Rate calculations. In the small network, the number of packets dropped is higher when the node density is lower; it decreases with the increase in node density. In the medium size network, this percentage is lower and is decreasing with the increase in node density. However, the decrease in the percentage of dropped packets for the medium network is not large.

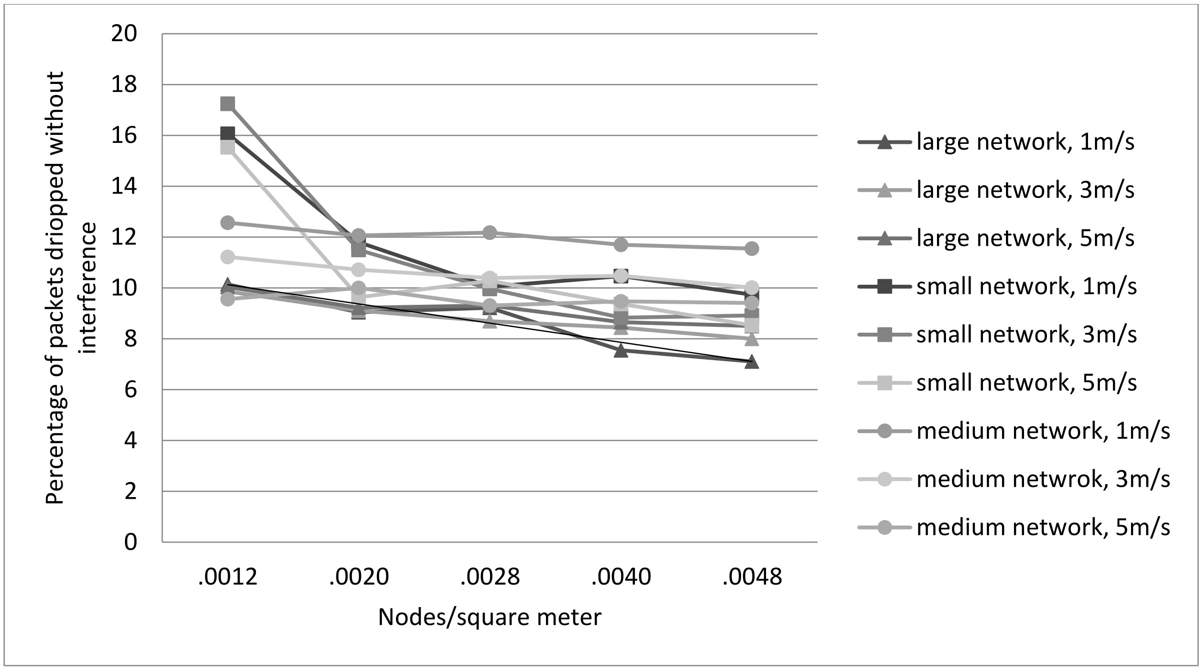


Figure 6.7: Percentage of packets in the network dropped due to thermal noise and Bit Error Rate calculations for all three network sizes.

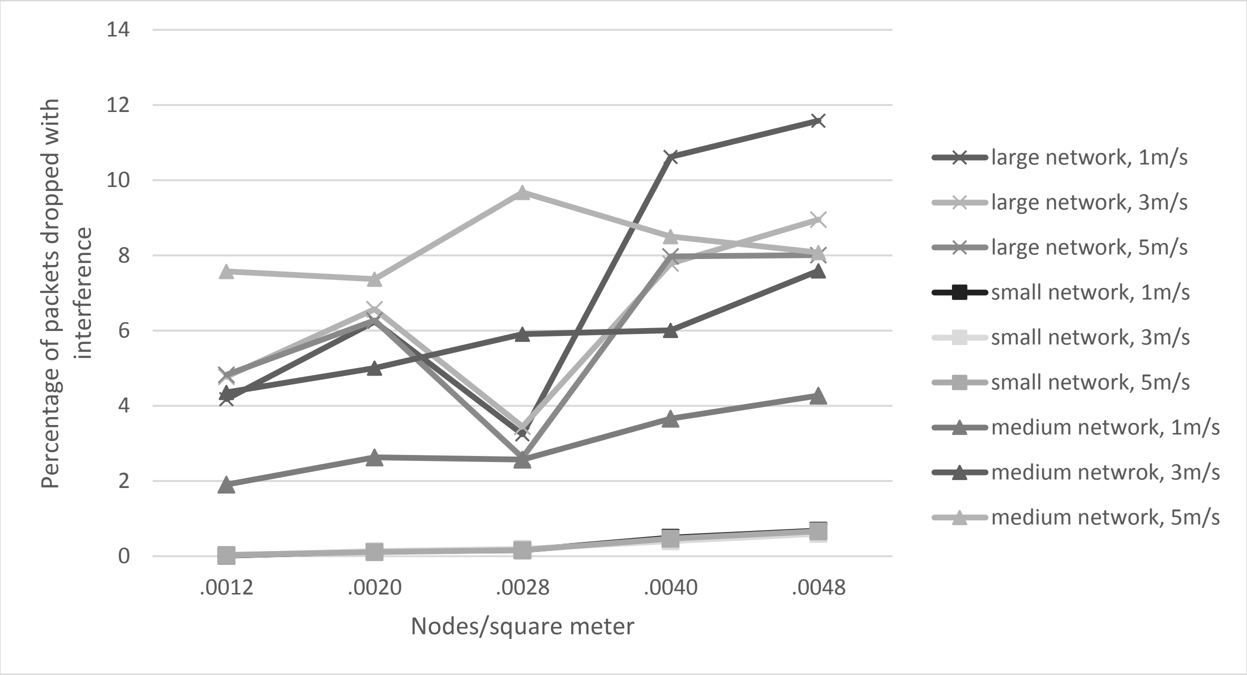


Figure 6.8: Percentage of the packets dropped with interference for all network sizes

The percentage of packets dropped in the network, due to other causes but interference, is not affected by the node density as much as it is affected by the network size, or in other words the number of hops required reaching the sink. If we compare between different network sizes with similar node densities, the nodes have on average the same number of neighbors. For larger networks, those neighbors might not have a direct connection to the sink as it is the case for the medium sized network.

Dropped packets due to interference however, are more affected by the node density and node speed as shown in figure 6.8. The large network suffers from more packet drops due to interference; it has the highest number of packets in the network. However, the main reason for packet drops at the radio layer is thermal noise. The interference in the medium sized network causes an increasing percentage of dropped packets with the increase in node density. Packet drops due to interference are affected by node speed; the slower networks are still performing slightly better than the faster networks.

The sparse small sized networks suffered from a higher number of packets that were dropped, because the received packets were lower in power than the receiver's sensitivity. This was in fact the main reason for the packet drops at the radio layer in the small sized networks with node densities 0.0012 and 0.002 nodes/m^2.

6.1.3 Packet Drops at the Routing Layer

A deeper look into the reason why some packets are dropped at the routing layer is taken. The packets are dropped due to different reasons as shown in the following figures.

Figure 6.9 shows that for the all mobile scenario, a big number of packets are dropped from the routing layer when the client node is busy; it means that the client node has too many packets to send. Dropping a packet at the routing layer means that the packet is not going to be retransmitted again, as it is the case when no acknowledgment is received; it means that the packet is permanently lost. This high number of packet drops is the major reason of packet loss for the medium sized networks.

It can be seen that on average up to 85% of the packets are dropped without being sent in the large network, and around 45% in the medium sized network. In the large sized network, when the node density is 0.0028 nodes/m^2, the number of packets dropped when the client is busy is lower. In the large sized network, with less node density, the network is too sparse and the nodes try to retransmit to the parent and hence they get busy. Moreover, in the large sized network with a larger number of nodes, the interference and the MAC layer back offs is the main cause for filling up the client nodes with packets to be sent.

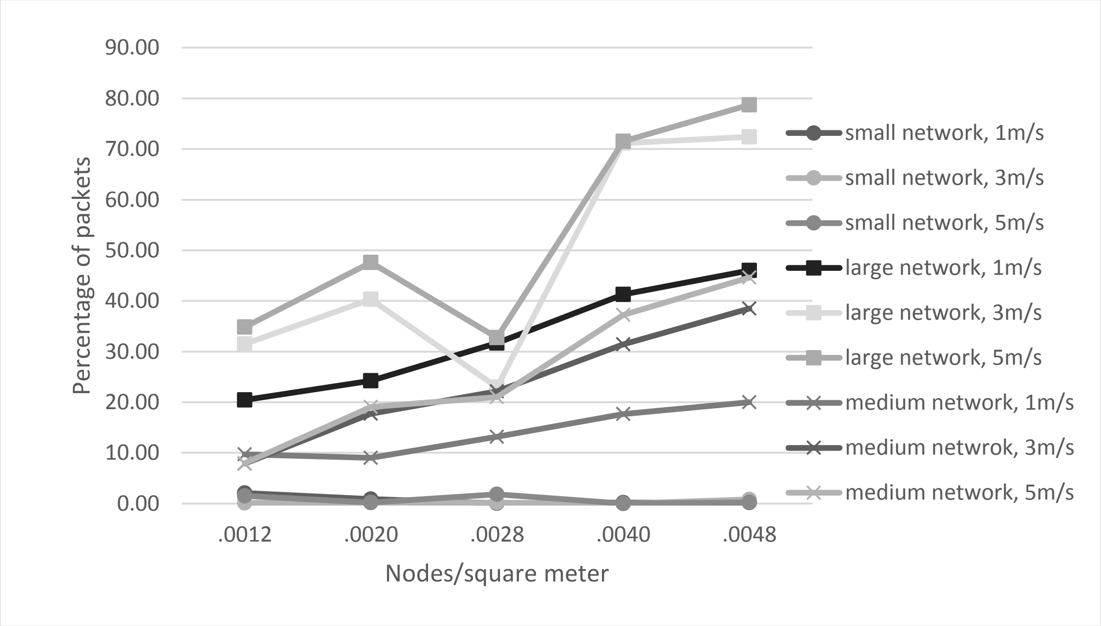


Figure 6.9: Average number of packets dropped due to busy client.

Figure 6.10 shows the average number of packets dropped due to lost acknowledgements or lost packets. It shows that for the lower density network, there is higher number of packets dropped due to reaching the maximum number of retransmissions. The client gets busy trying to send or resend unacknowledged packets and more packets are dropped.

There is a difference in the network's performance with change in node speed; slower speeds don't cause as much packet drops as higher speeds, because the nodes have more time staying within reach of each other.

Figure 6.10 shows as well that the number of retransmissions for the networks with node density 0.0028 nodes/m^2 has an increase when compared to the other scenarios with different node densities. The same scenario shows a decrease in the number of packets dropped due to interference. The denser networks suffer from interference and hence dropped packets at the radio layer and buffer's overflowing. The sparse networks loose a higher number of packets below receiver's sensitivity and hence suffer as well from buffer overflow.

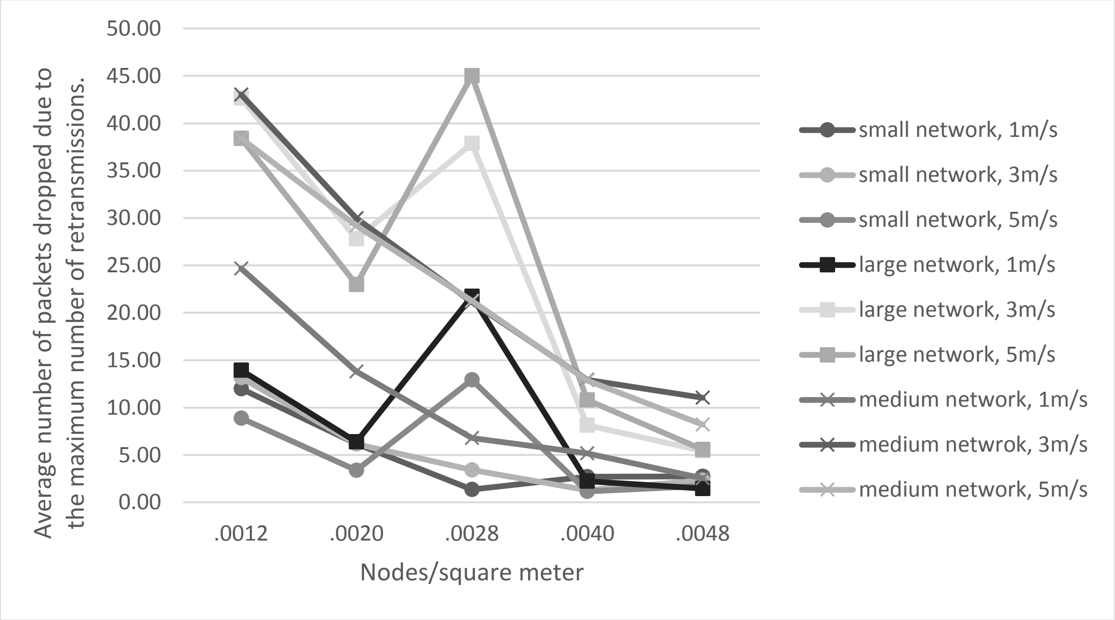


Figure 6.10: Average number of packets dropped when reaching the maximum number of retransmissions for all network sizes.

6.1.4 The MAC Layer

The next layer to look at is the MAC layer, and the number of congestions in the network. Which means that when there are more packets, to send or resend, there are more packets traversing the network, and this causes collisions at the MAC layer, when two nodes send a packet at the same time, and hence more lost packets.

However, since the MAC layer is not part of this research, we will only investigate the effect of the current layer and the collisions happenings. The starting node density, for each of these scenarios, is constant due to the node placement with a uniform distribution; the mobility pattern will affect node distribution which is not uniform once the nodes start moving. There can be situations where there is a high concentration of nodes in a small area, and this causes collisions when all nodes try to send at the same time. This means that, even at the network layer, there is a problem to use the channel and send the packets.

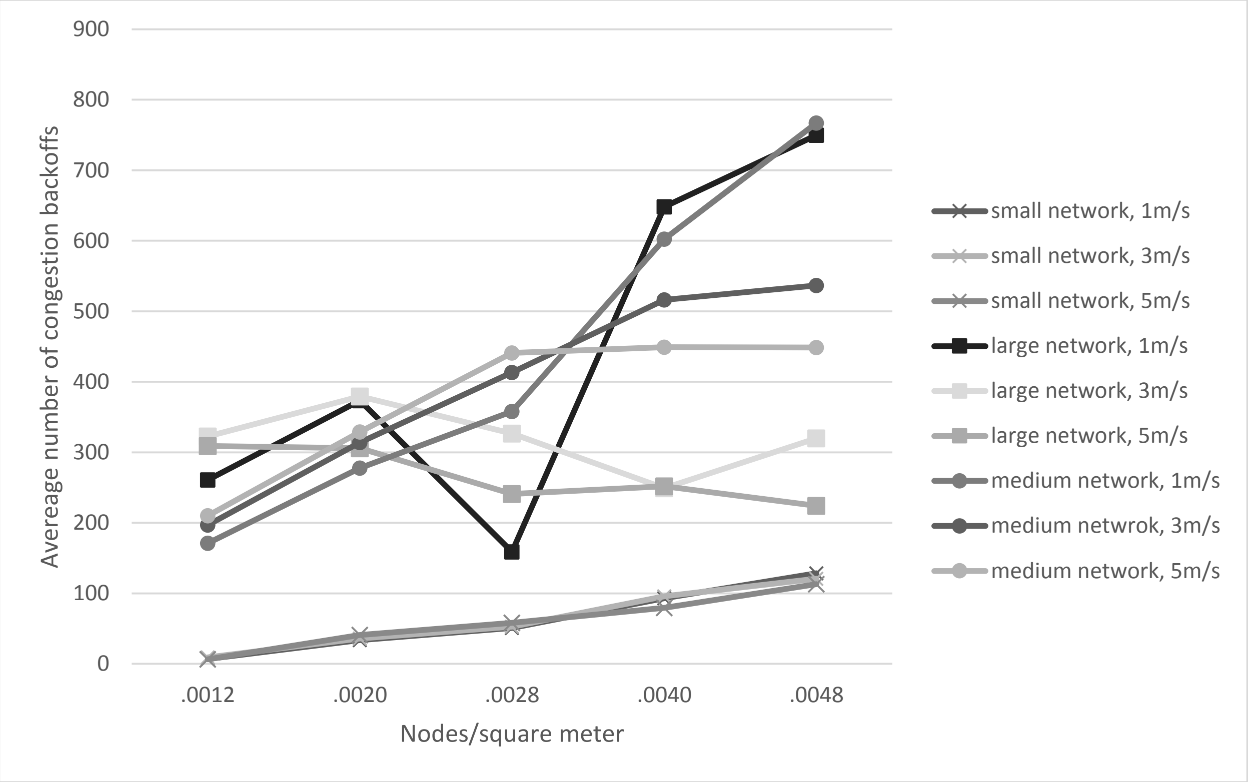


Figure 6.11: Average number of congestion back-offs in the network.

The stats of the MAC layer can answer the question about the channel availability; the number of times the channel is not clear for sending, and the node’s need to start a timer to attempt another transmission.

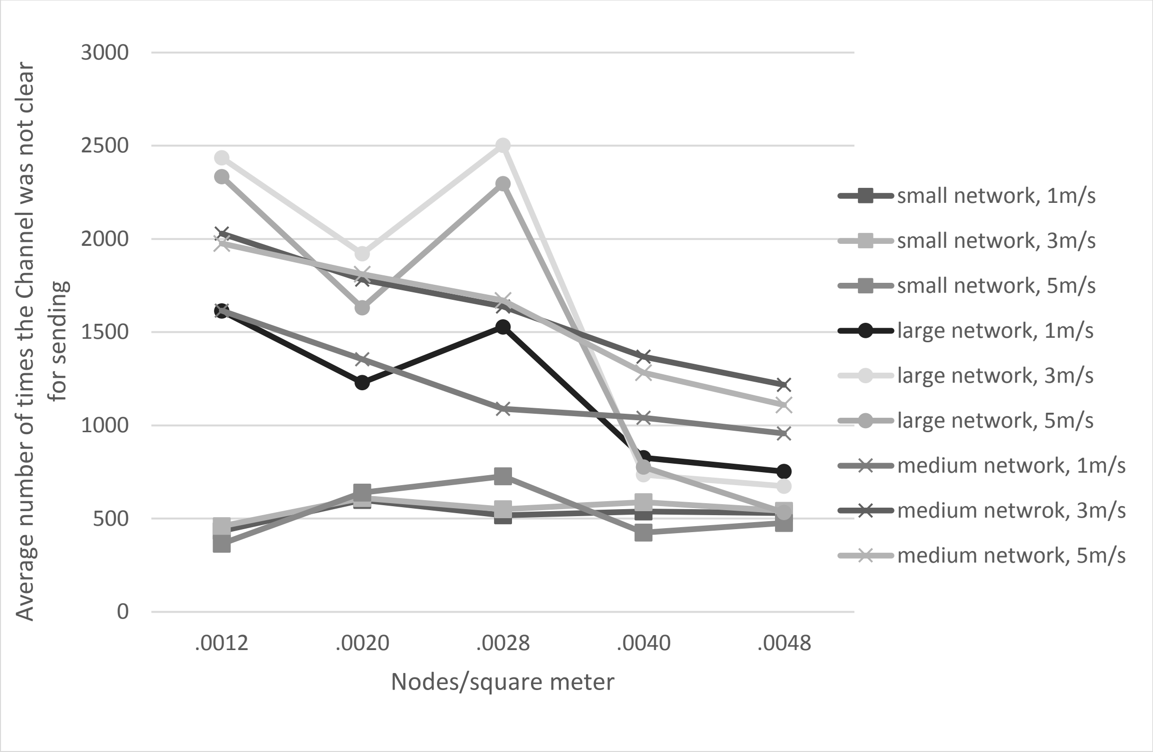


Figure 6.12: Average number of times the channel was not clear for sending.

6.1.5 Network Overhead.

There are two types of overhead in this network, the control overhead due to beacon transmission, and the retransmissions of packets due to lost packets or lost acknowledgments.

Figure 6.13 shows the percentage of packets that are retransmitted to the overall transmitted packets in the network. It can be seen that this percentage decreases with the increase in node density. The spike that is seen for the small network with high speed is related to the direct connection to the sink.

The results from sections 6.1.1 to 6.1.5 show that increasing the size of the network increases the number of packets dropped in the network. While mobility didn't affect the PRR of the small size network, it heavily affected the large sized network.

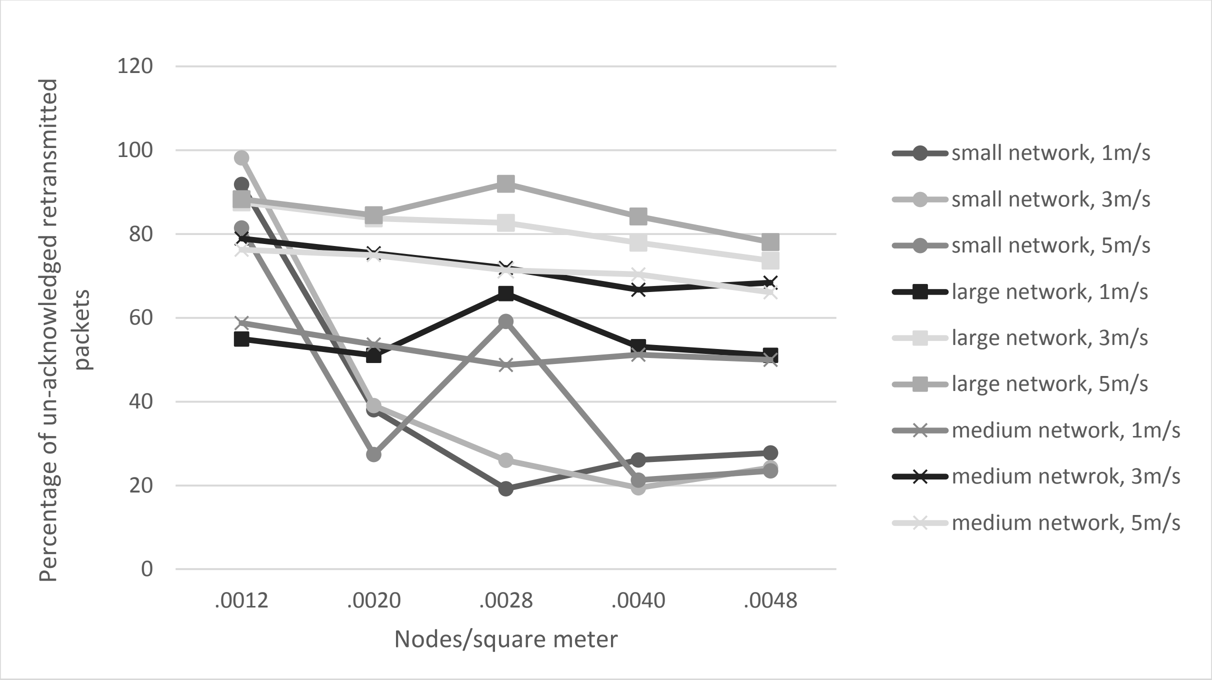


Figure 6.13: Percentage of packets that are not acknowledged and hence retransmitted.

The all mobile scenarios showed that the large sized network had the lowest performance. Therefore, the large sized network was chosen to investigate the performance of the network when running FNA-CTP with the interest in achieving an increase in performance.

6.2 Mixed mobile-static network scenarios

This section is dedicated to the study of FNA-CTP performance sensitivity to node density and ratio of fixed-to-mobile nodes in the network. The simulation parameters are shown in Table 6.2.

Table 6.2: the simulation scenario parameters

Parameters	**Unit/Value**
Field parameters	
Network size	150 by 150 meters^2
Number of Nodes	27, 45, 63, 90, 108 nodes
Sink position	(75,150)

Application layer parameters	
Data traffic	0.333 packets/second
Fixed nodes	
Number of fixed odes	
Number of fixed Nodes for network with 45 nodes	3x4, 4x4, 4x5, 5x5, 5x6
Number of fixed Nodes for network with 63 nodes	4x4, 4x5,5x5, 5x6,6x6
Number of fixed Nodes for network with 90 nodes	5x5, 5x6, 6x7, 7x7, 7x8
Number of fixed Nodes for network with 108 nodes	5x6, 6x7, 7x7, 8x8, 8x9
Number of retransmissions for fixed nodes	20
Minimum to maximum beacon interval for fixed nodes	64 ms to 250s
Sent cache size for fixed nodes	8
FE queue size for fixed nodes	24
Transmission power of fixed nodes	-3, -1, 0 dbm
Mobile Nodes	
Beacon interval for the mobile nodes	15 seconds
Number of retransmissions for mobile nodes	10
Transmission power of mobile nodes	-3bm
Speed of mobile nodes	1, 3, 5 meters/s

The results of these simulations will be presented in the following sections.

6.2.1 PRR

Figure 6.14 shows the PRR received at the sink with the network of 45 nodes. It can be seen that, for low node density, we are not receiving a PRR higher than 50% unless more than 50% of the nodes are static. Moreover, there is a clear advantage when the static nodes have higher transmission power.

When the node density increases, as shown in figures 6.15-6.17, we can see a trend that less static nodes are required to increase the performance of the network. In fact, for the network with 108 nodes, with already 28% of static nodes, the performance increases to about 55% for the case with low mobility and high transmission power for the static nodes.

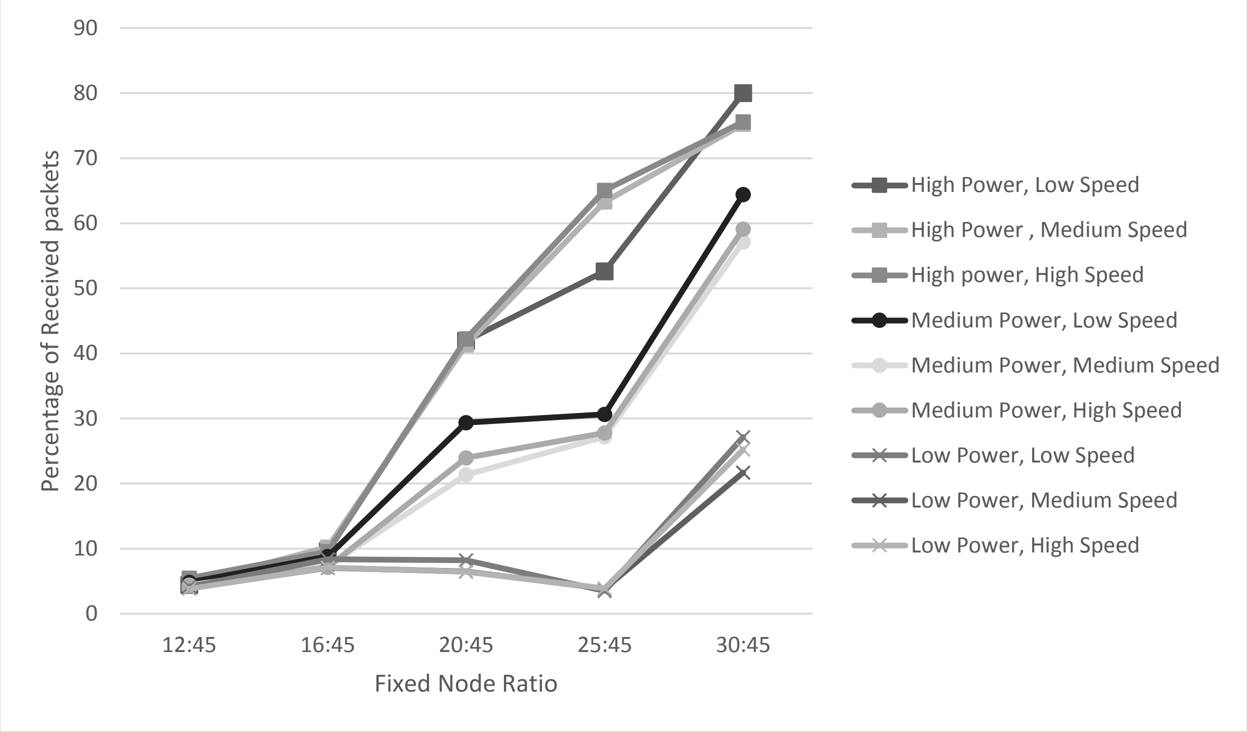


Figure 6.14 PRR of the network with 45 with different FNR values

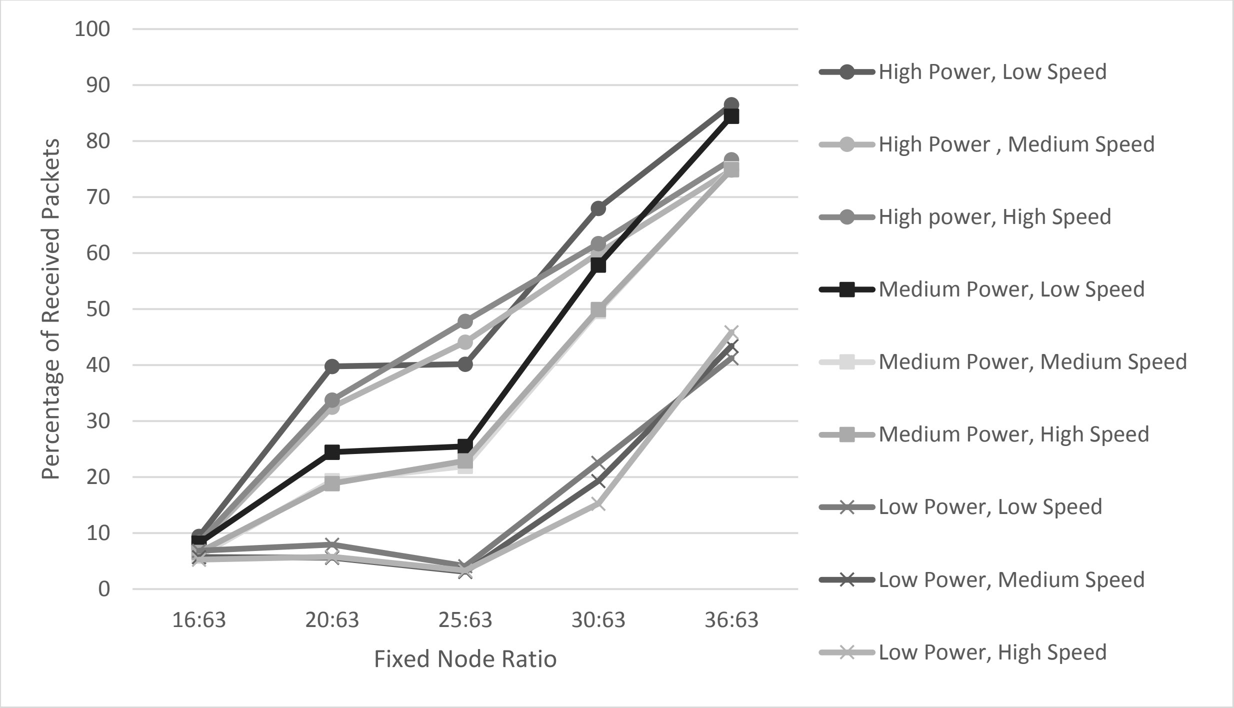


Figure 6.15: PRR of the network with 63 with different FNR values

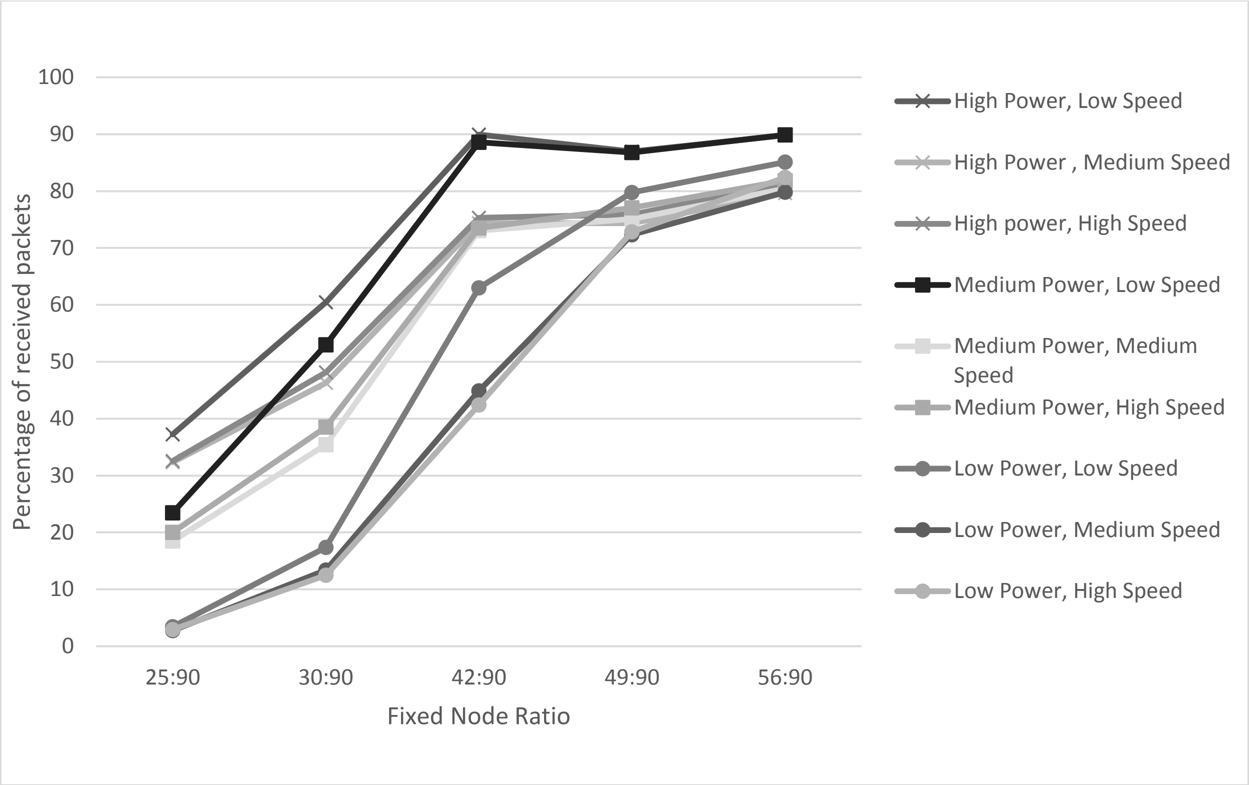


Figure 6.16: PRR for the network with 90 nodes with different FNR values

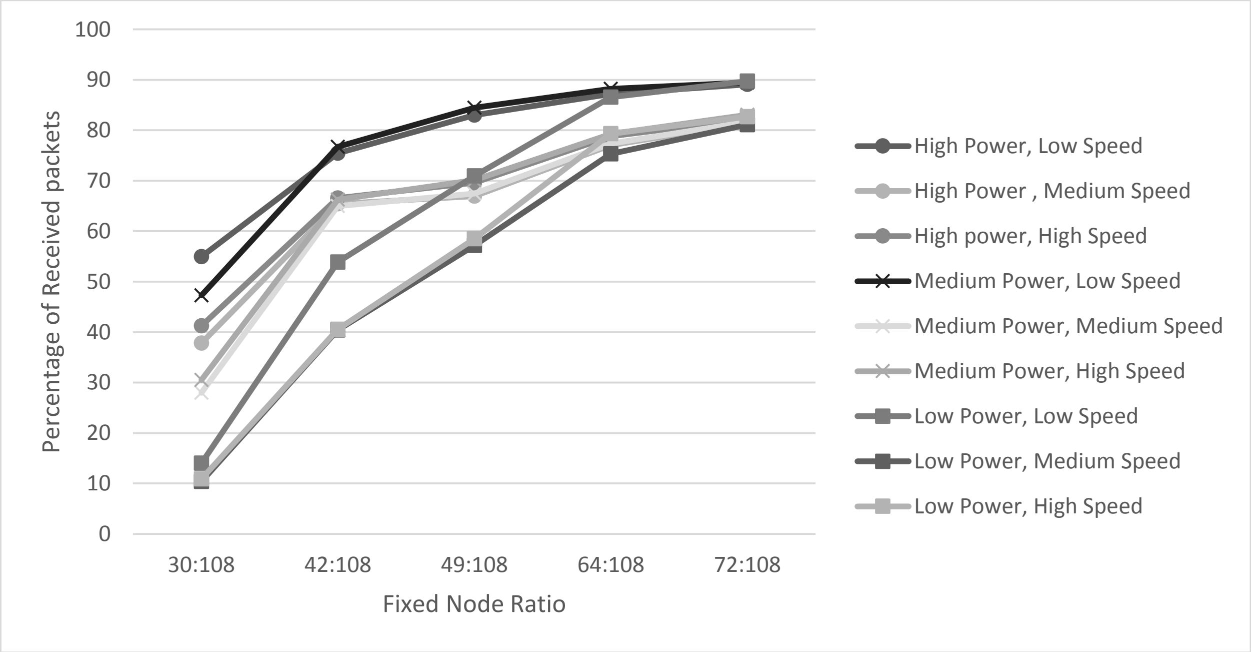


Figure 6.17: PRR of the network with 108 nodes with different FNR values

Another observation is the convergence of the PRR values for the 108 node network once more than 60% of the nodes are static. For all different values of transmission power for the static nodes, the performance increases. This trend starts showing at the 63 node network.

6.2.2 The Radio Layer

Following figures show the number of packets dropped, in the radio layer, due to thermal noise and due to interference. It is important to distinguish between the dropped packets, because a packet can be dropped due to interference but it is still getting retransmitted again by the routing layer. However, if an application packet is dropped it will not be retransmitted again.

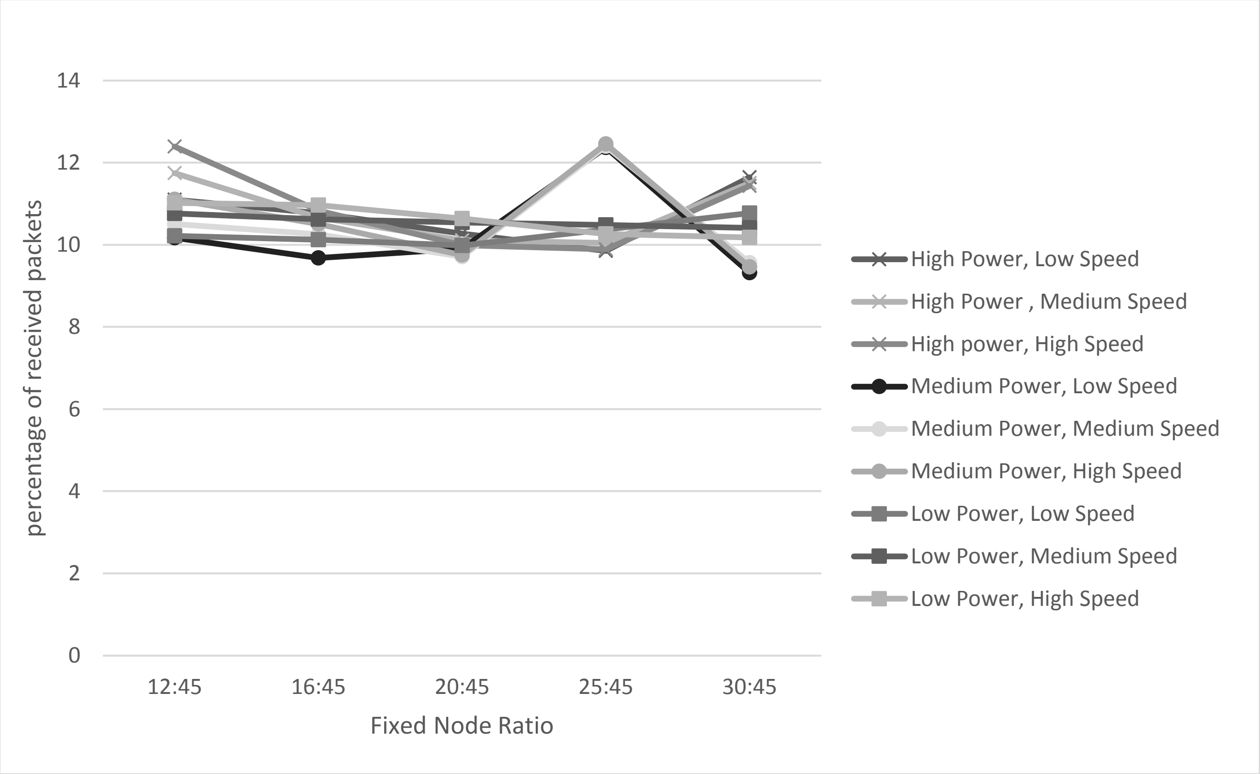


Figure 6.17: Average number of packets dropped in the network due to thermal noise for the 45 node network.

On average, the percentage of packets dropped in the sparser networks from BER is within 2% range. However, there is an increase in the interference with the increase in the number of static nodes and their power, which peaks at about 55% of static nodes; this is due to the non-uniform distribution of nodes and the tendency of the mobile nodes to move towards the center of the network. This explains the peak in interference in the rest of the networks as well. With the increase in the number of nodes and the increase in the power of the static nodes, there is more interference between the sensor nodes. Overall in this case, the low density network is the one with the lowest interference, especially if the static nodes have a transmitting power of -3dbm.

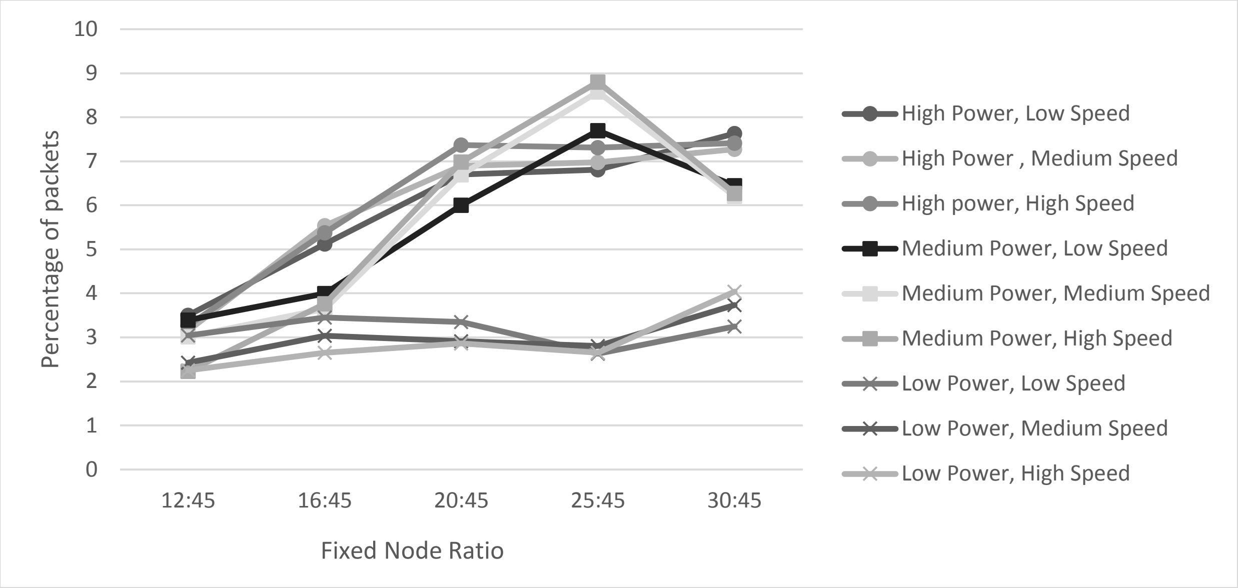


Figure 6.18: Average number of packets dropped in the network due to interference in the 45 node network

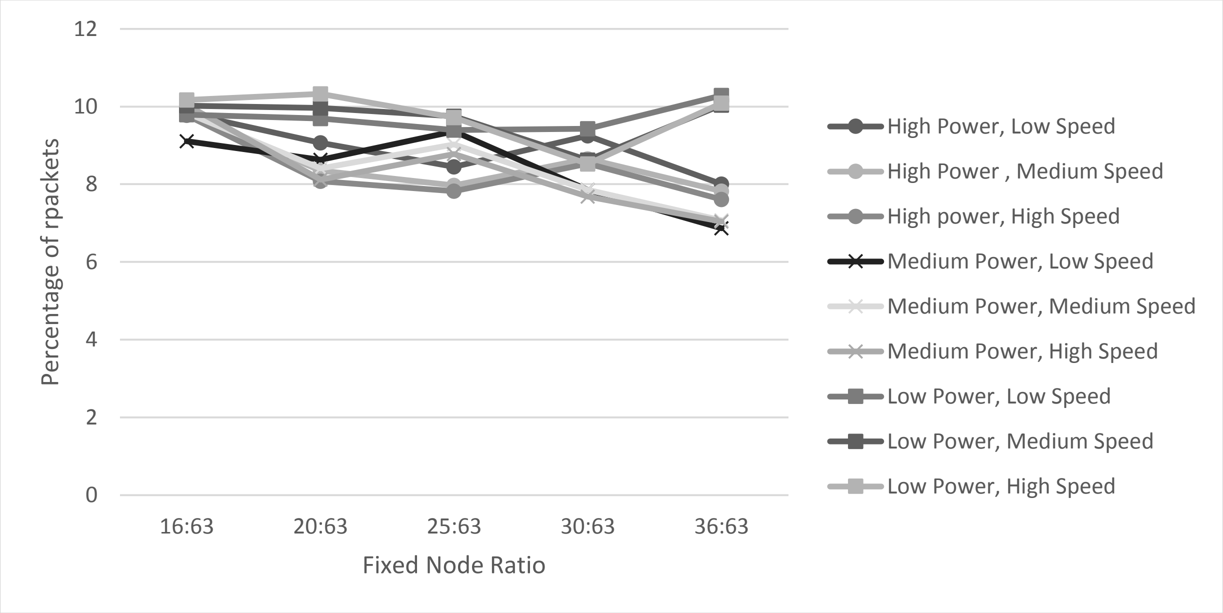


Figure 6.19: Average number of packets dropped at the radio layer due to noise for the network with 63 nodes.

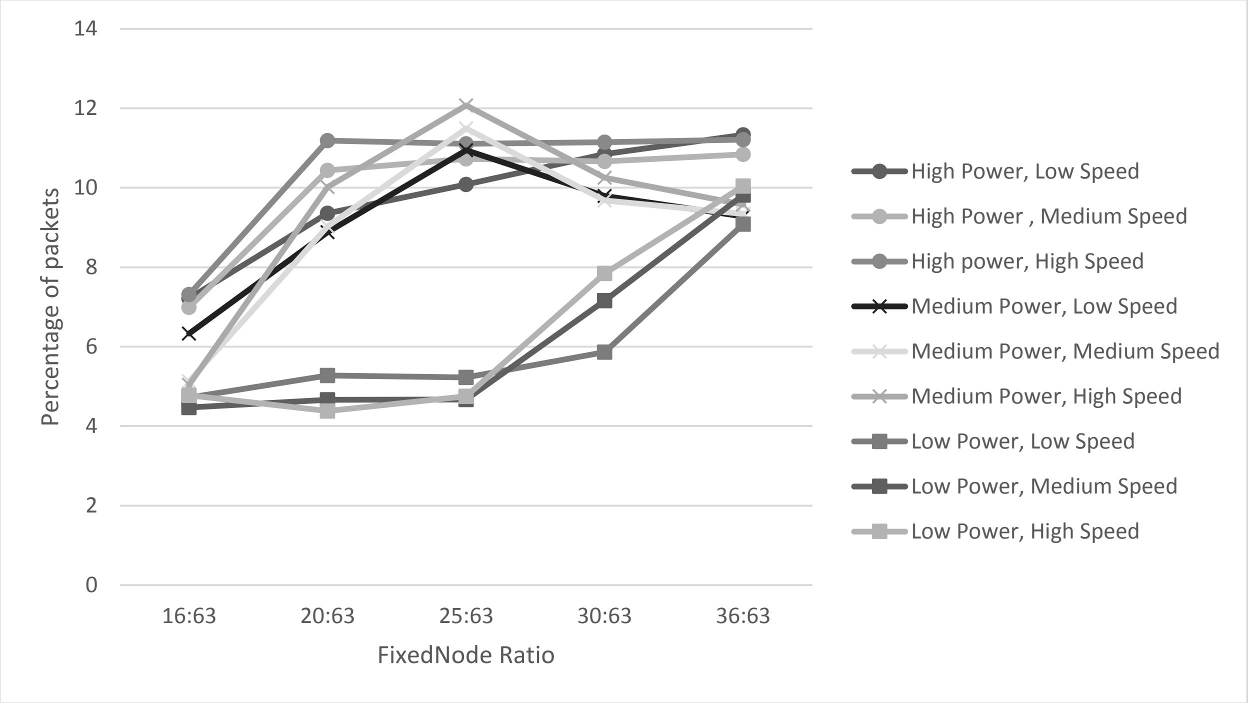


Figure 6.20: Average of packets dropped in the network due to interference in the 63 node network

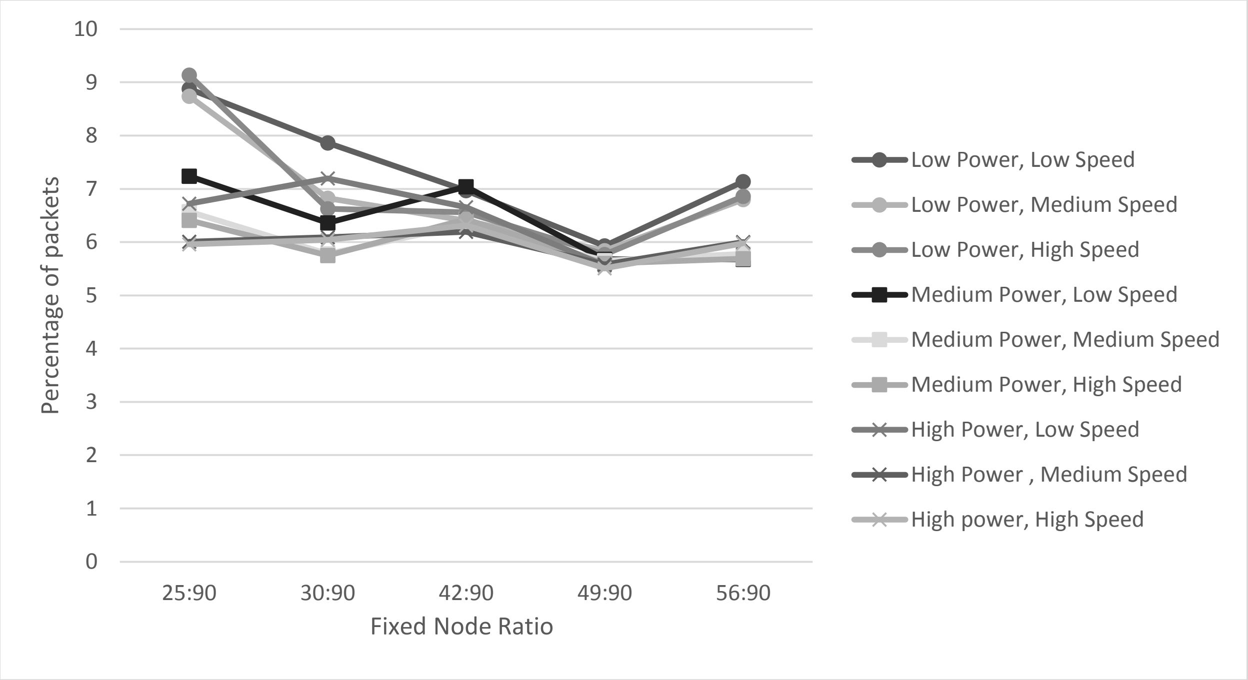


Figure 6.21: Average number of packets dropped due thermal noise for the 90 node network

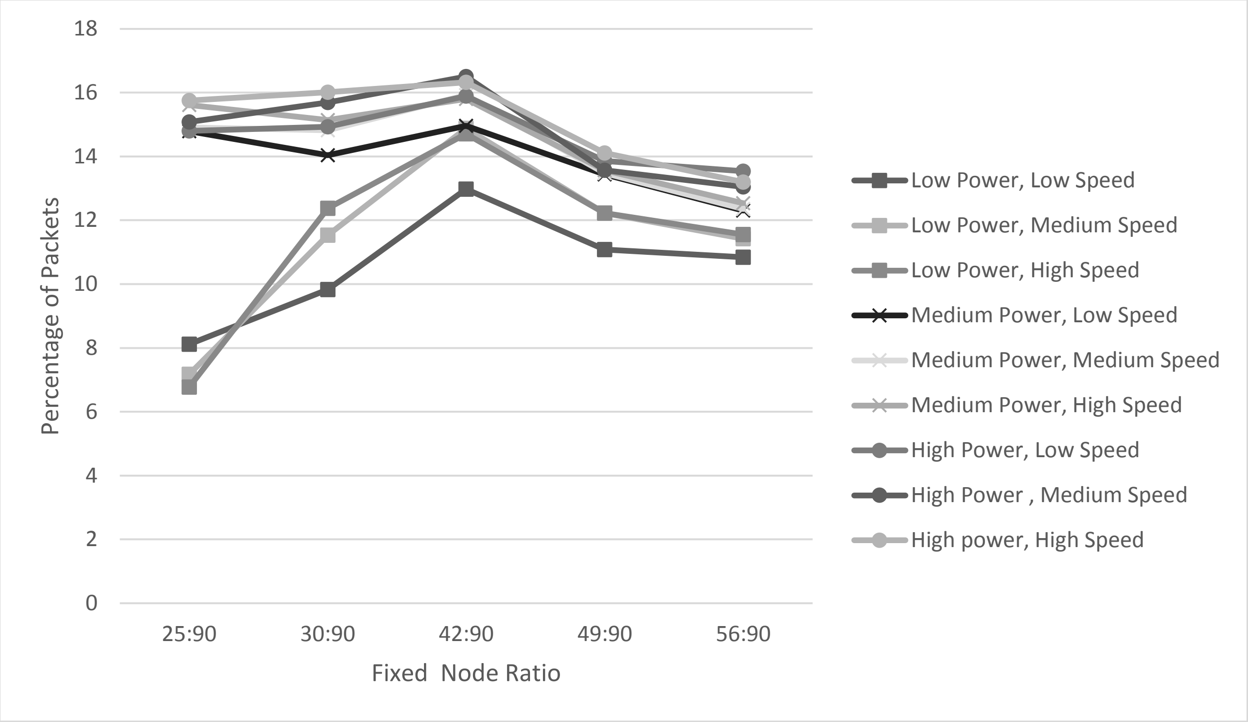


Figure 6.22: Average number of packets dropped in the network due to interference for the 90 node network.

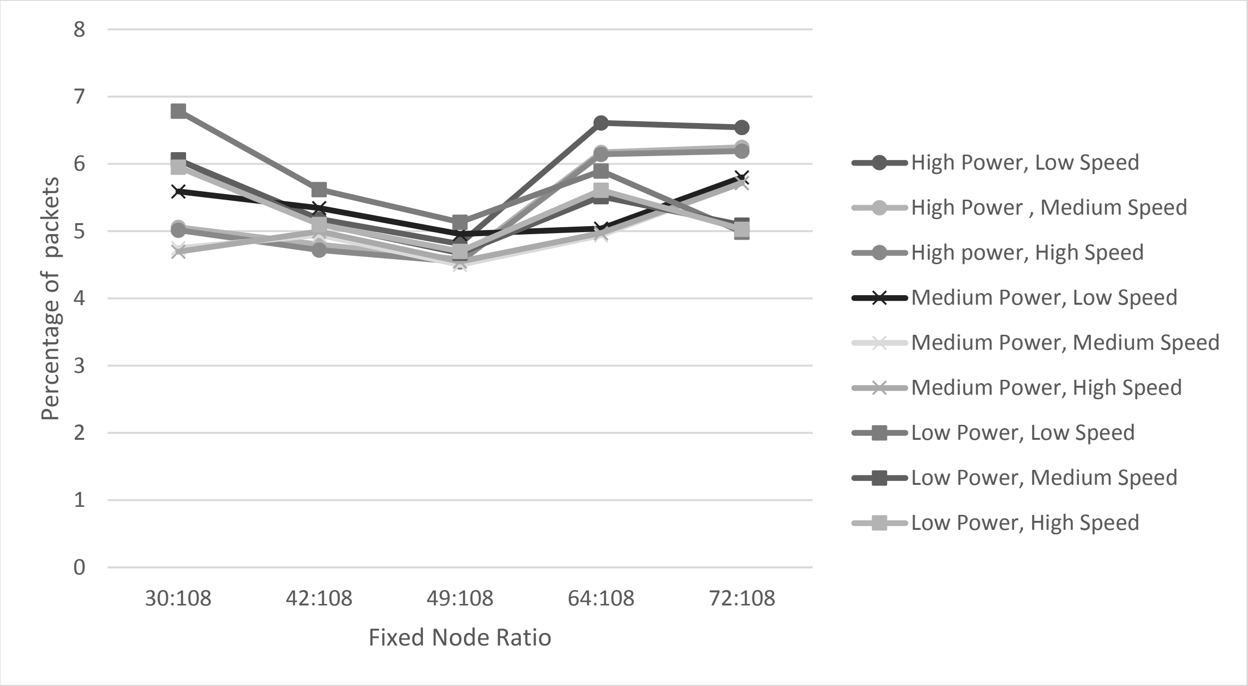


Figure 6.23: Average number of packets dropped in the radio layer due to noise in the network with 108 nodes.

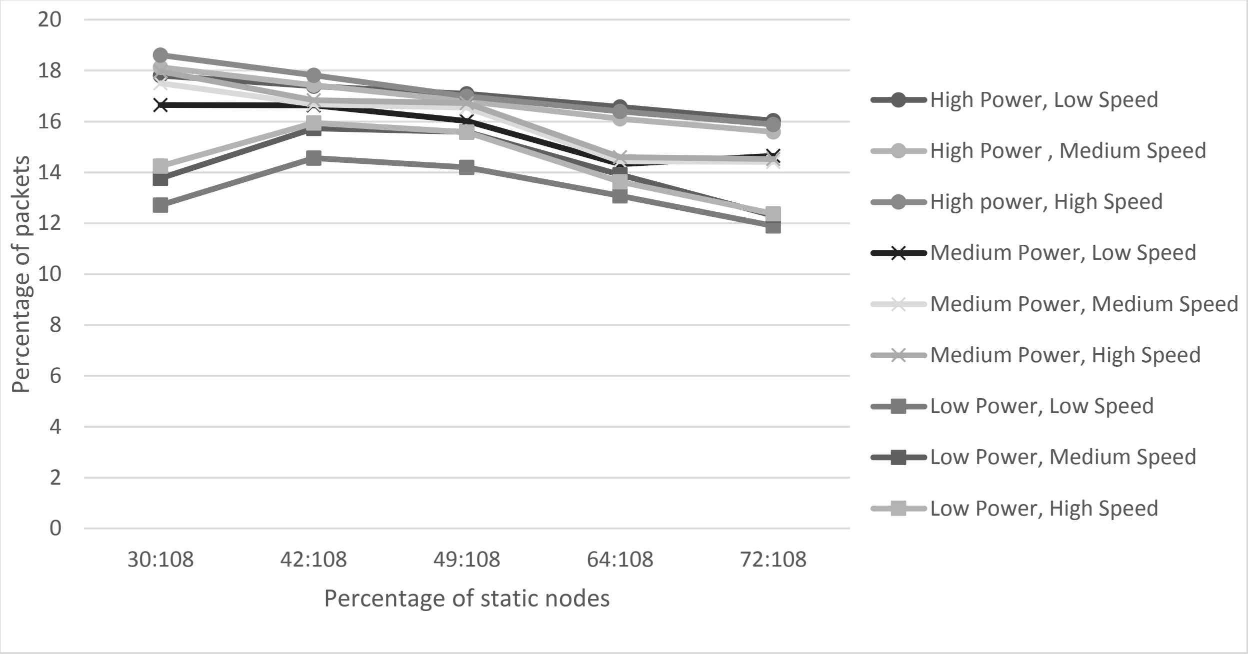


Figure 6.24: Average number of packets dropped in the network due to interference for the 108 node network.

6.2.3 The Routing Layer

Figures 6.25-6.29 show the dropped packets due to reaching the maximum number of retransmissions. In the 108 node network it can be seen that the average number of dropped packets starts converging to below 2 packets on average per node.

When the network has a less nodes, there is variation in the temporal network topology, especially when the static nodes are low. In all network scenarios the number of packets dropped due to reaching the maximum number of retransmissions drops in the network with the highest FNR value.

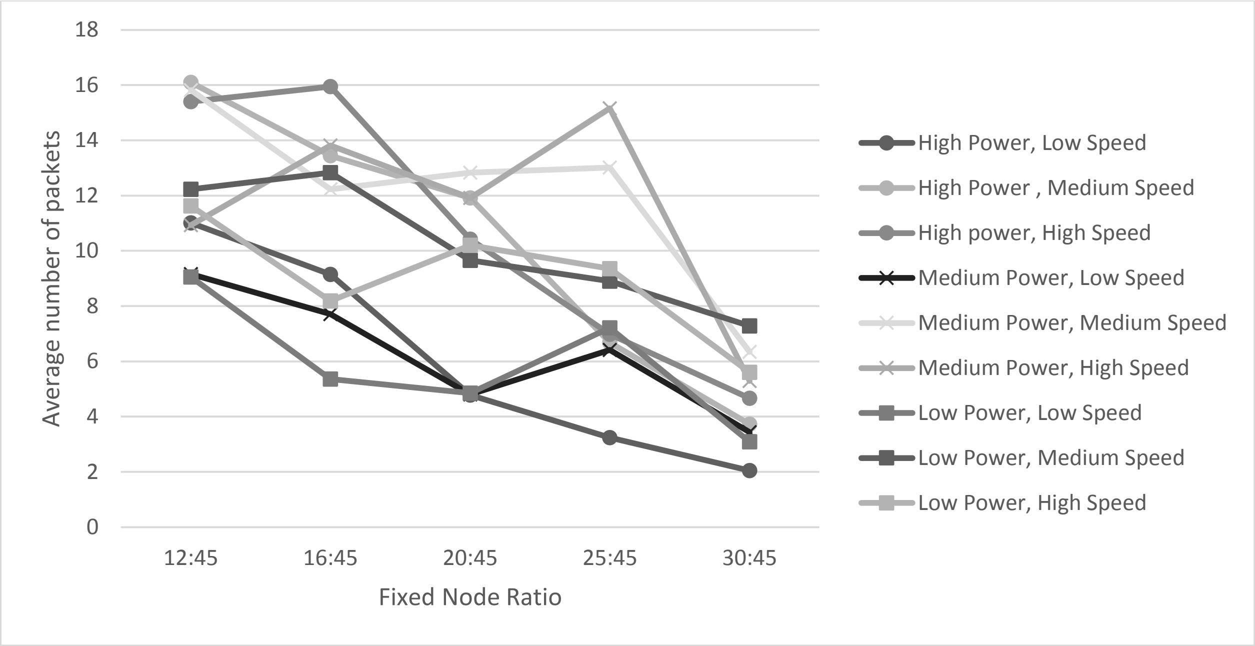


Figure 6.25 Average number of dropped packets per node due to the maximum number of retransmissions for the 45 node network

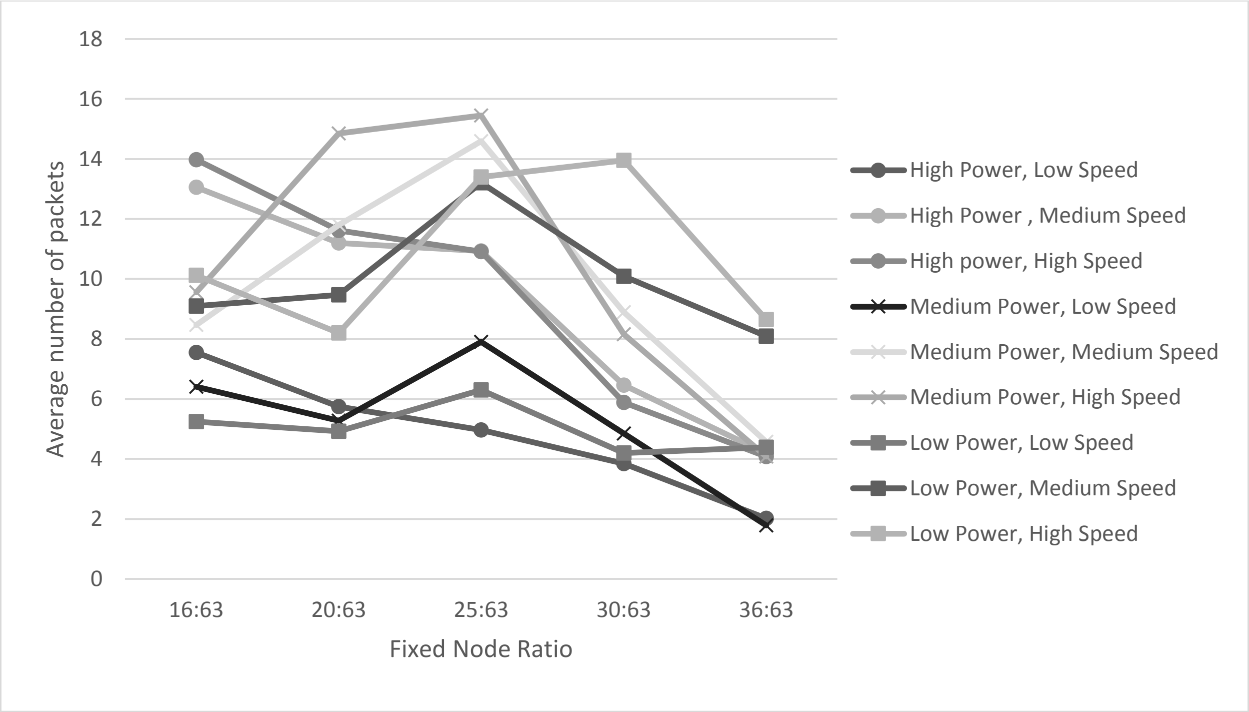


Figure 6.26: Average number of dropped packets due to the maximum number of retransmissions for the 63 node network.

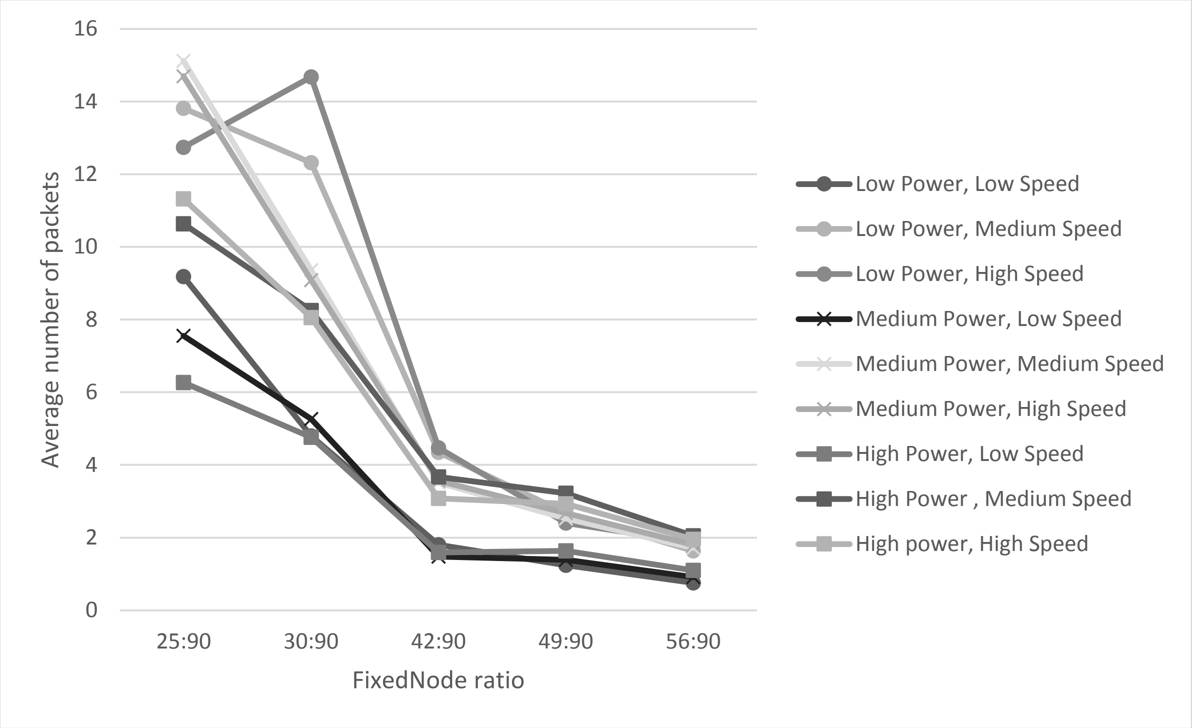


Figure 6.27: Average number of packets dropped after reaching the maximum number of retransmissions in the 90 node network.

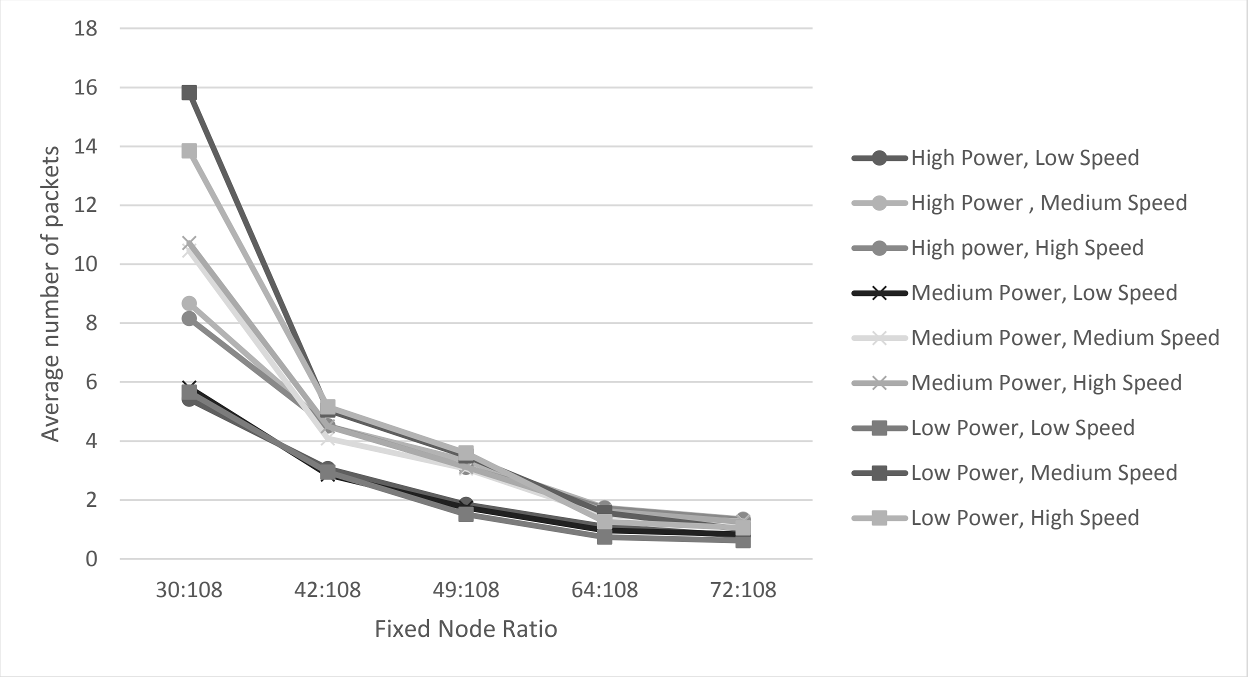


Figure 6.28: Average number of dropped packets due to the maximum number of retransmissions for the 108 node network.

Figures 6.29-6.32 show the packets dropped because the client node was busy. This means the packets are permanently dropped. It is observed that for the 45 node network, a high number of packets get dropped, especially in the network with low transmission power. Therefore, there is an increase in loosing packets below sensitivity; there are more attempts to resend the packet.

When the static nodes constitute more than 56% of the overall nodes, then the number of dropped packets gets below 20 packets on average per node.

The network with 108 nodes had the lowest number of dropped packets, even if only a 28% of the nodes are static and the rest of the nodes move with low speed.

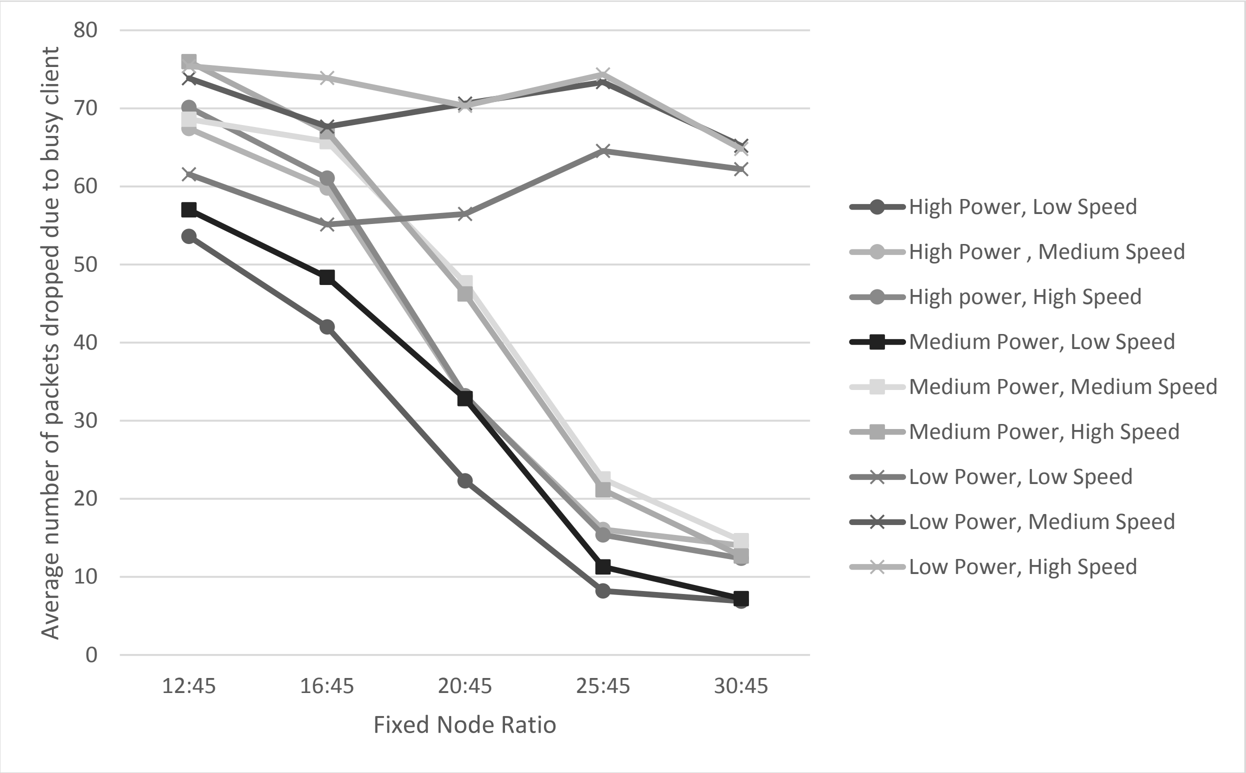


Figure 6.29: Average number of packets dropped due to a busy client at the application layer for the 45 node network.

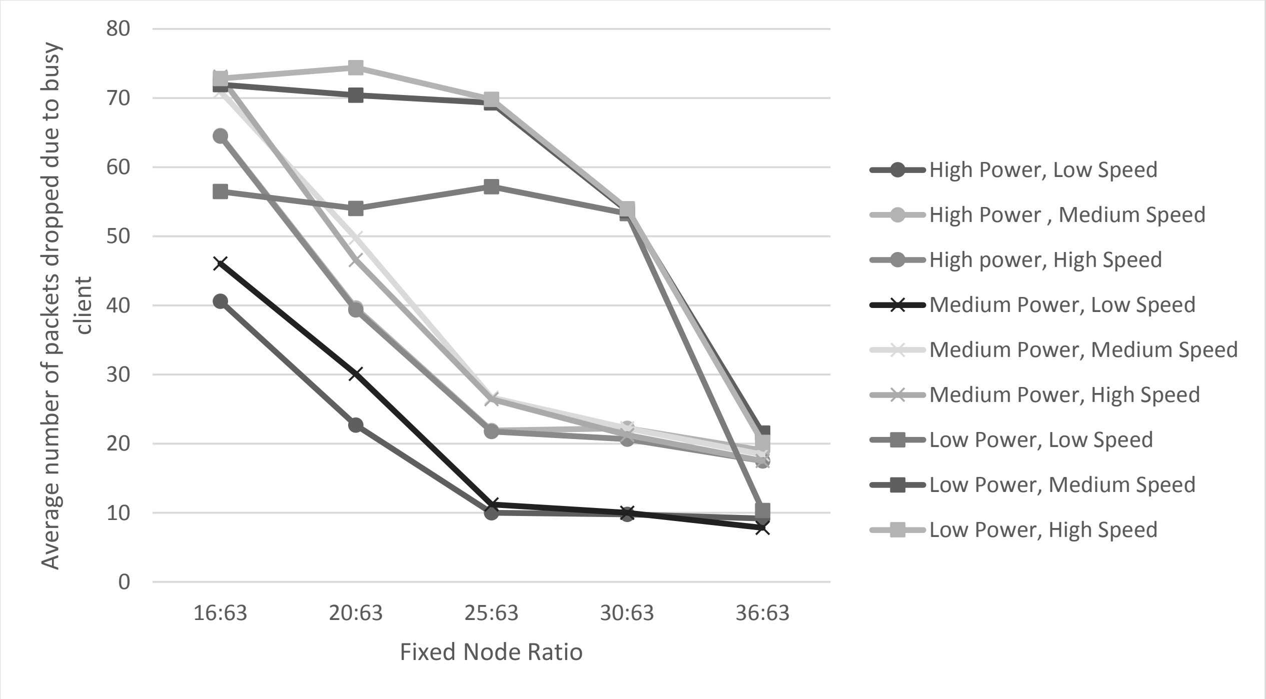


Figure 6.30: Average number of packets dropped due to the busy client for the 63 node network.

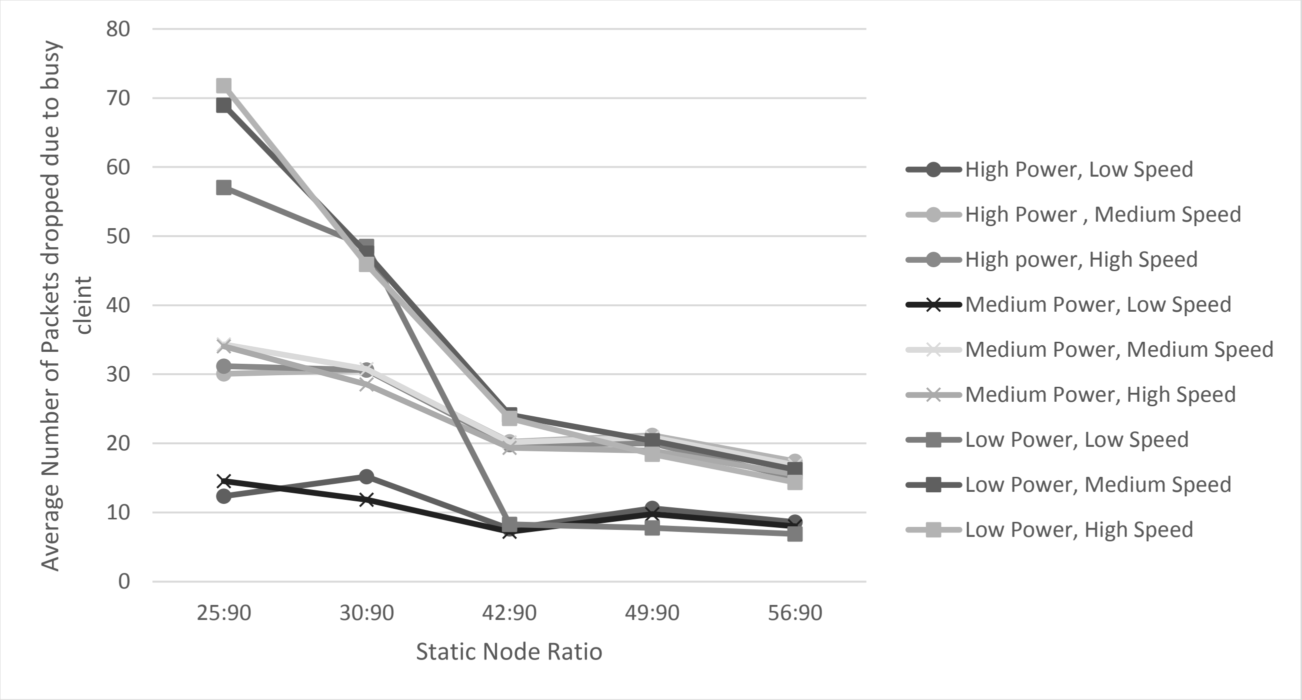


Figure 6.31: Average number of packets dropped due to the busy client in the 90 node network.

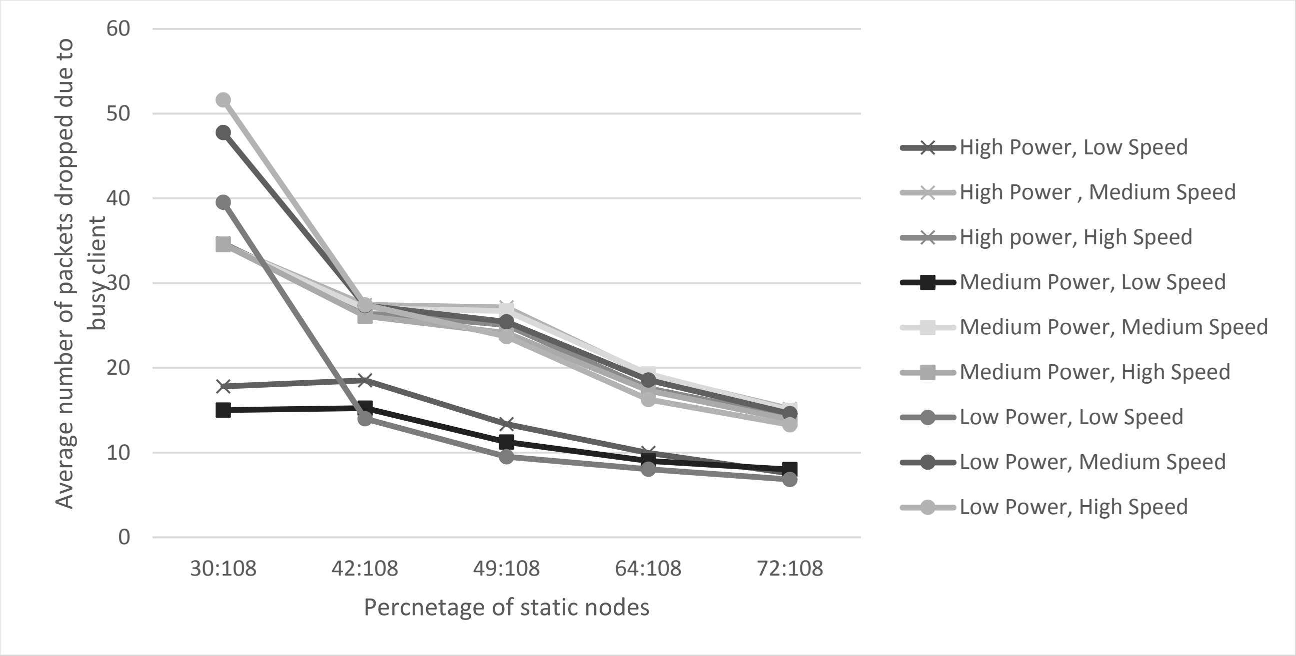


Figure 6.32 Average number of application packets dropped per node 108 node network due to the busy client.

6.3 Networks with bigger field sizes

After looking into the large sized network with different node densities, we choose to simulate a larger sized network. In this step, a node density of 0.004 nodes/m^2 is used, due to its better performance in terms of PRR and lower interference than the network with higher density. The parameters are shown in tables 6.3 and 6.4.

Table 6.3: the network parameters for the 200 by 200 network

Parameters	**Unit/Value**
Field parameters	
Network size	200 by 200 meter^2
Number of nodes	160 nodes
Sink position	Top center of the field
Application layer parameters	
Data traffic	0.333 packets/second
Fixed nodes	

Number of fixed nodes	6x7, 8x8, 9x9, 9x10
Number of retransmissions for fixed nodes	20
Minimum to maximum beacon interval for fixed nodes	64ms to 250s
Sent cache size for fixed nodes	8
FE queue size for fixed nodes	24
Transmission power of fixed nodes	-3, -1, 0 dbm
Mobile Nodes	
Beacon interval for the mobile nodes	15 seconds
Number of retransmissions for mobile nodes	5
Transmission power of mobile nodes	-3bm
Speed of mobile nodes	1, 3, 5 meter/s

Table 6.4: the parameters of the 250 by 250 sized network

Parameters	**Unit/Value**
Field parameters	
Network size	250 by 250 meter^2
Number of nodes	250 nodes
Sink position	Top center of the field
Application layer parameters	
Data traffic	0.333 packets/second
Fixed Nodes	
Number of fixed nodes	8x8, 8x9, 9x10, 11x11, 12x12
Number of retransmissions for fixed nodes	20
Minimum to maximum beacon interval for fixed nodes	64ms to 250s
Sent cache size for fixed nodes	8
FE queue size for fixed nodes	24
Transmission power of fixed nodes	-3, -1, 0 dbm

Mobile Nodes	
Beacon interval for the mobile nodes	15 seconds
Number of retransmissions for mobile nodes	5
Transmission power of mobile nodes	-3bm
Speed of mobile nodes	1, 3, 5 meter/s

The simulation results of both network scenarios are provided in sections 6.4.1 to 6.4.3. We investigate the PRR of the larger networks when using the node density that has performed well in section 6.2. The goal is to investigate the effect of increasing the network size on the performance.

6.3.1 PRR

The PRR results of both network sizes are shown in figures 6.33 and 6.34. In the 200 by 200 network, a PRR of more than 80% can be achieved with 50% static nodes.

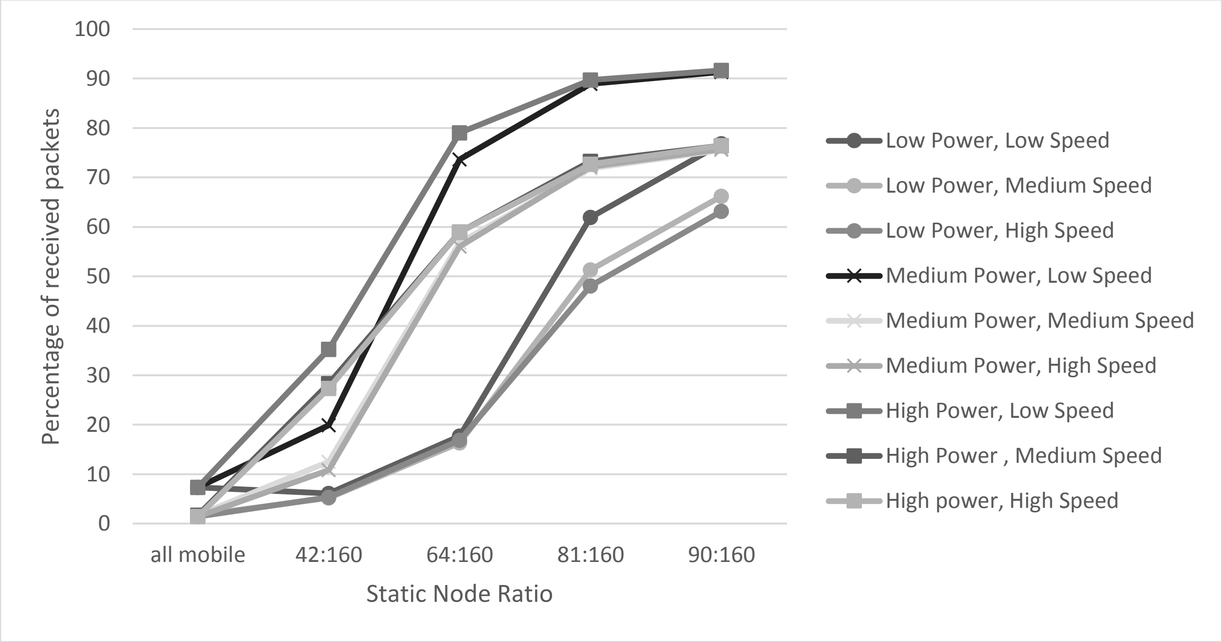


Figure 6.33: Average number of received packets at the sink for the 200 by 200 network.

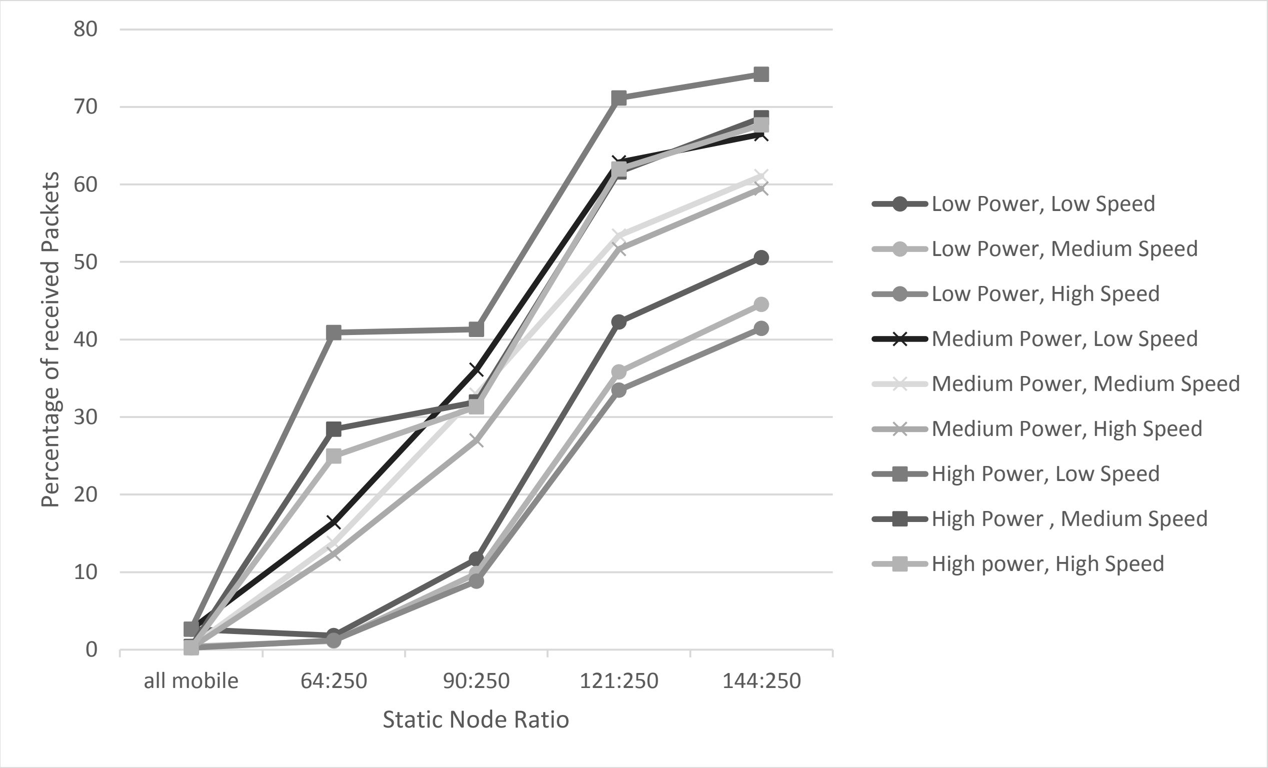


Figure 6.34: Average number of received packets at the sink for the 250 by 250 network.

The 250 by 250 network had a lower PRR. It doesn't reach 80%. Which means that the chosen node density of 0.004 nodes/m^2 is not sufficient to ensure high performance. The important factors will be discussed in further detail in section 6.4

6.4 Results Discussion and Analysis

6.4.1 Connectivity

As mentioned in Ch.2, the connectivity is a measure that we can use to ensure a fully connected network. In order to investigate the relation between the current network scenarios and connectivity, we investigate the nodal degree for the 90 node network. Based on Eq.1 the minimum node degree to ensure a connected network should be 10.117.

Looking into figure 6.35, we can see the average node degree taken over time in the network. These values were taken as snapshots of the network topology, and the connections are based on the distance between the nodes. Further on, figure 6.35

shows the static to static node degree and average node degree with PRR values shown in the background.

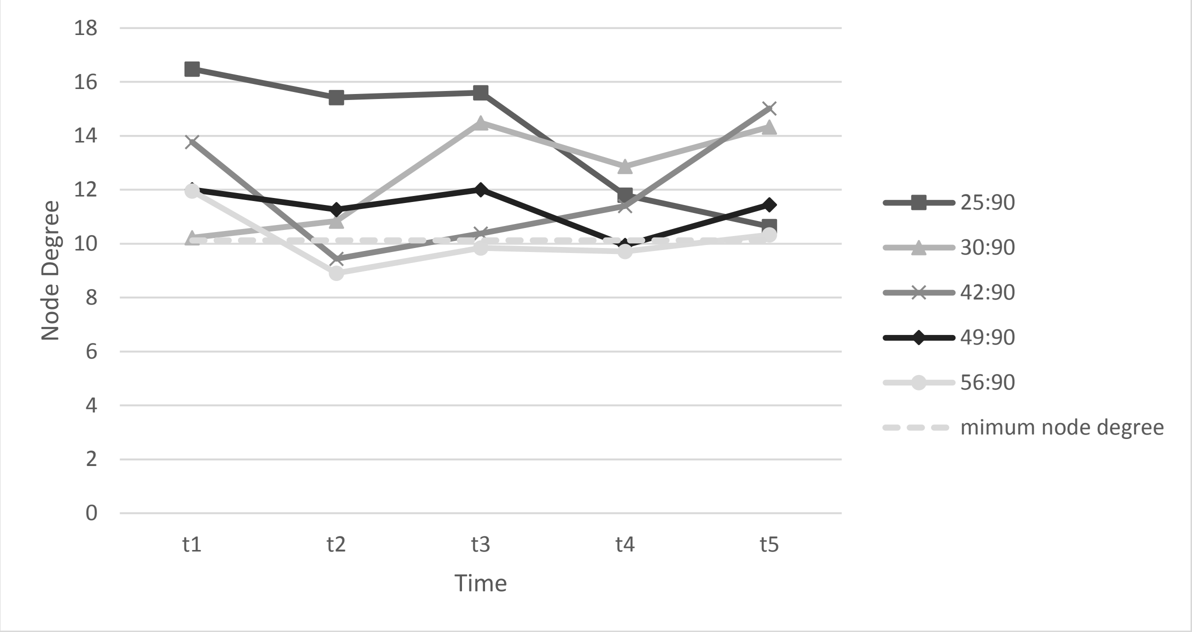


Figure 6.35: Node degree of the network at different points in time in the 90 node network

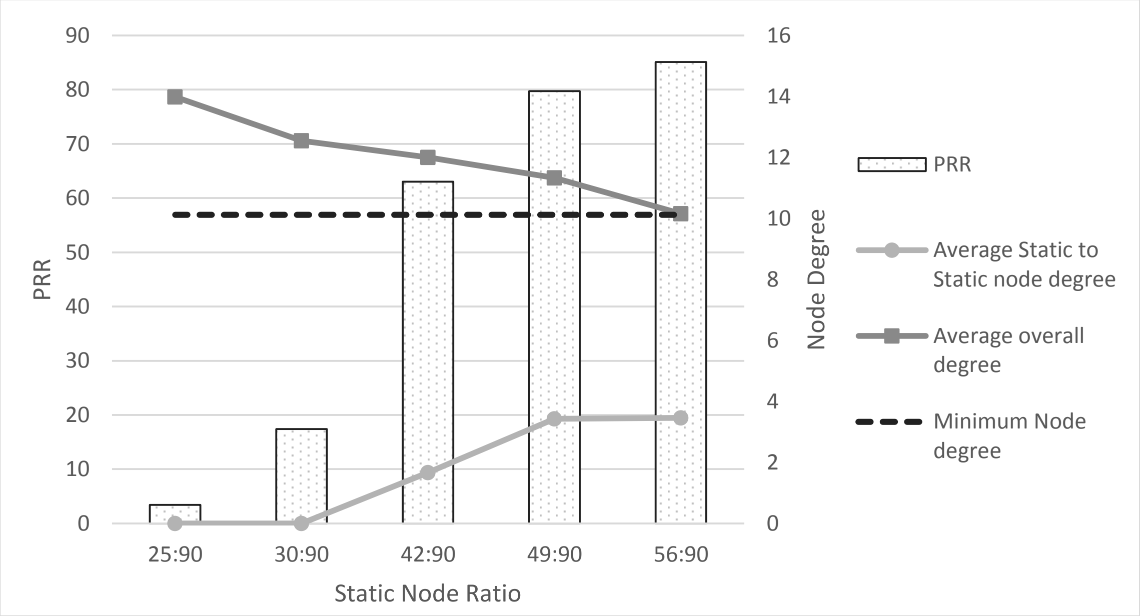


Figure 6.36: Comparison between the PRR and the average node degree and the static to static node degree in the 90 node network.

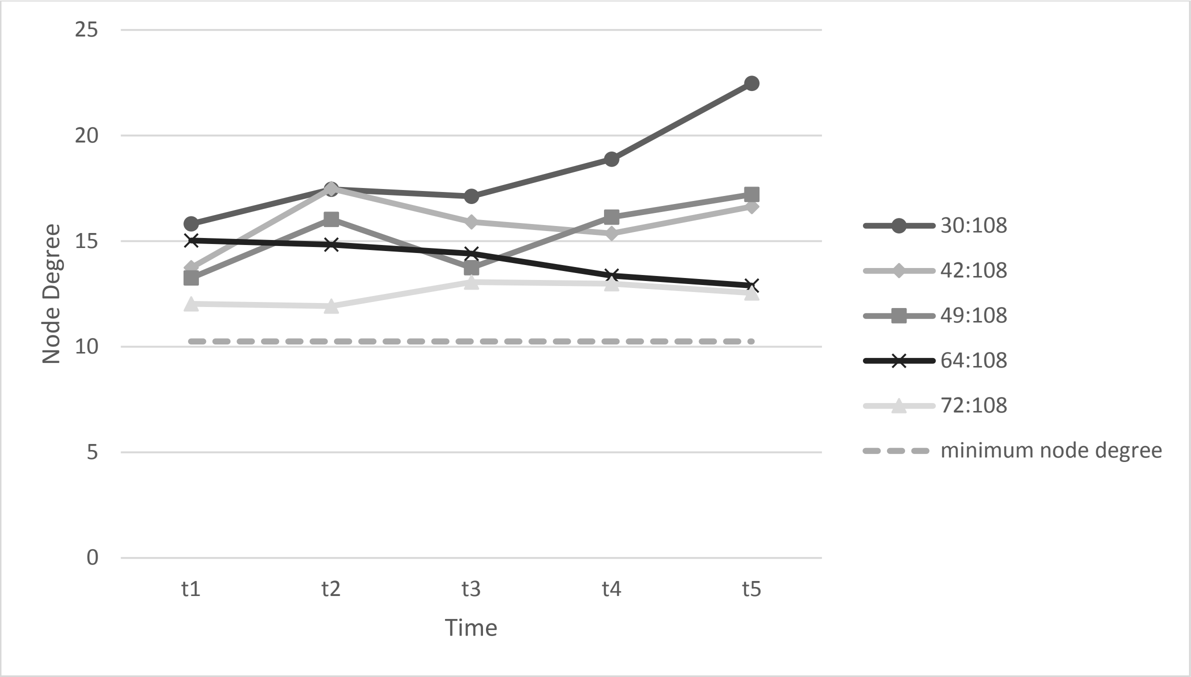


Figure 6.37: The average node degree of the network with 108 nodes over time

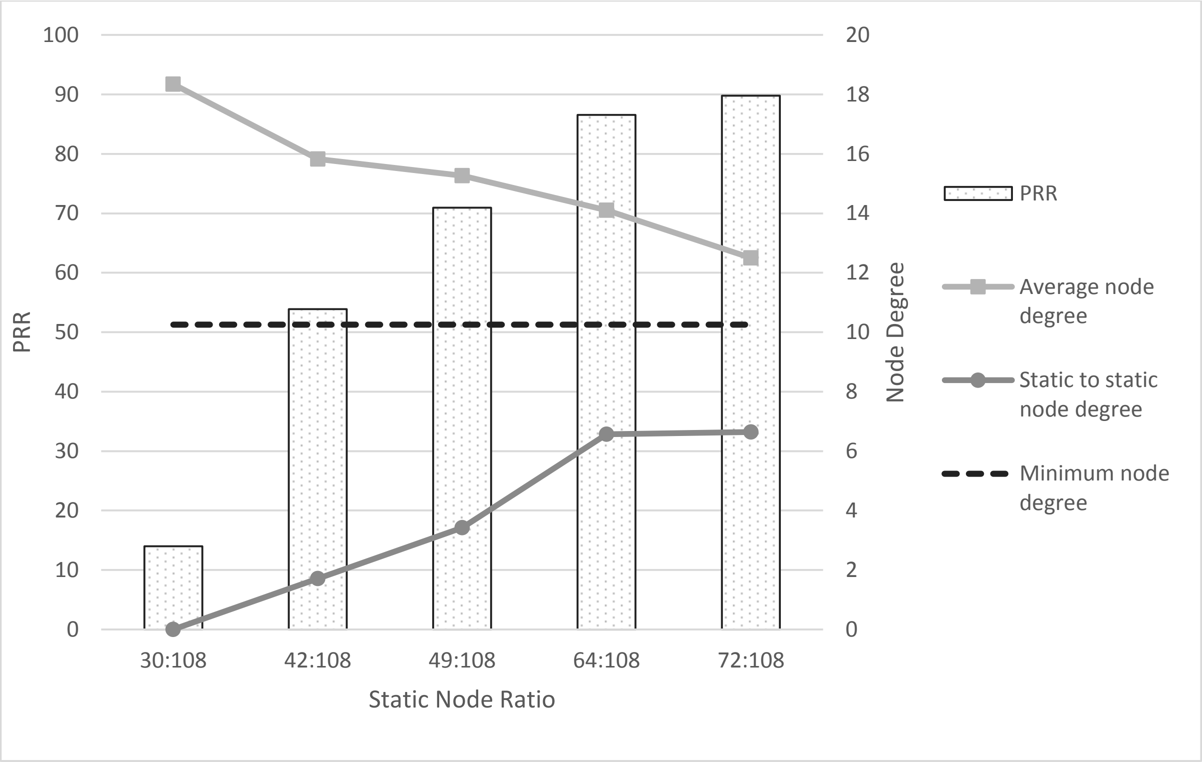


Figure 6.38: PRR, average node degree, static to static node degree and minimum required node degree in the 108 network.

In both cases, the overall node degree was higher than the minimum required node degree calculated from Eq.1. However, the PRR was not high in all cases. The static to static node degree was lower than the minimum required node degree as per Eq.1. However, this equation doesn't include transmission power or network size.

When looking into Eq.2, it relates node transmission range, field size with the number of nodes. In the previous two cases, the number of nodes varied, but the transmission range and the network size were fixed. Based on the second equation, the minimum number of nodes for the 150 by 150 network should be 180 sensor nodes. In the 90 node network we were able to reach a PRR above 80%.

Even though the average node degree was higher than the minimum node degree, the PRR in the network was still low when the static nodes were below 40% of the overall nodes. This means that a higher node degree and therefore network connectivity doesn't ensure a high PRR; the drop in PRR is not because of the number of neighbors a node has. High network connectivity doesn't mean a good performance. The static to static node degree affects the performance. The increase in the static to static node degree leads to an increase in the network's performance.

6.4.2 The Trickle Algorithm

The Trickle algorithm is an important feature of CTP in static scenarios. After the network start up phase, when all the nodes in the network have a path to the sink, unless they are totally out of range and isolated, there is no actual need to continuously send control beacon at short time intervals. However, when mobility is present, the trickle doesn't keep up with the frequent link breakages, especially when the nodes move in and out of a node's range very quickly. In FNA-CTP the trickle for the mobile nodes is turned off.

Another approach would be to turn off the trickle for all nodes in the network, and have the entire nodes send at fixed intervals. Sending too many beacons will cause a high overhead and interfere with the data packets in the network, and sending a few beacons will cause a lot of lost packets because of outdated parents that are not available anymore.

Here, we show briefly how the beacon interval affects the network's performance. We chose the network scenario of Ch.5 but with all nodes mobile and varied the beacon interval. The PRR of this network is shown in figure 6.39.

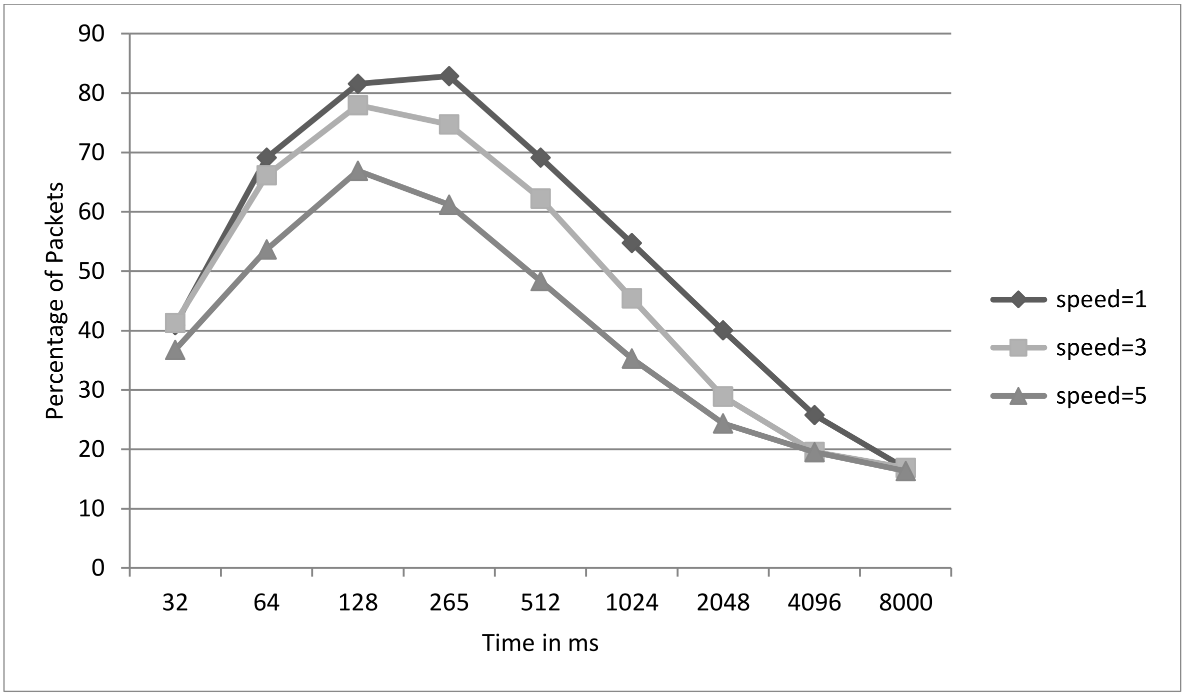
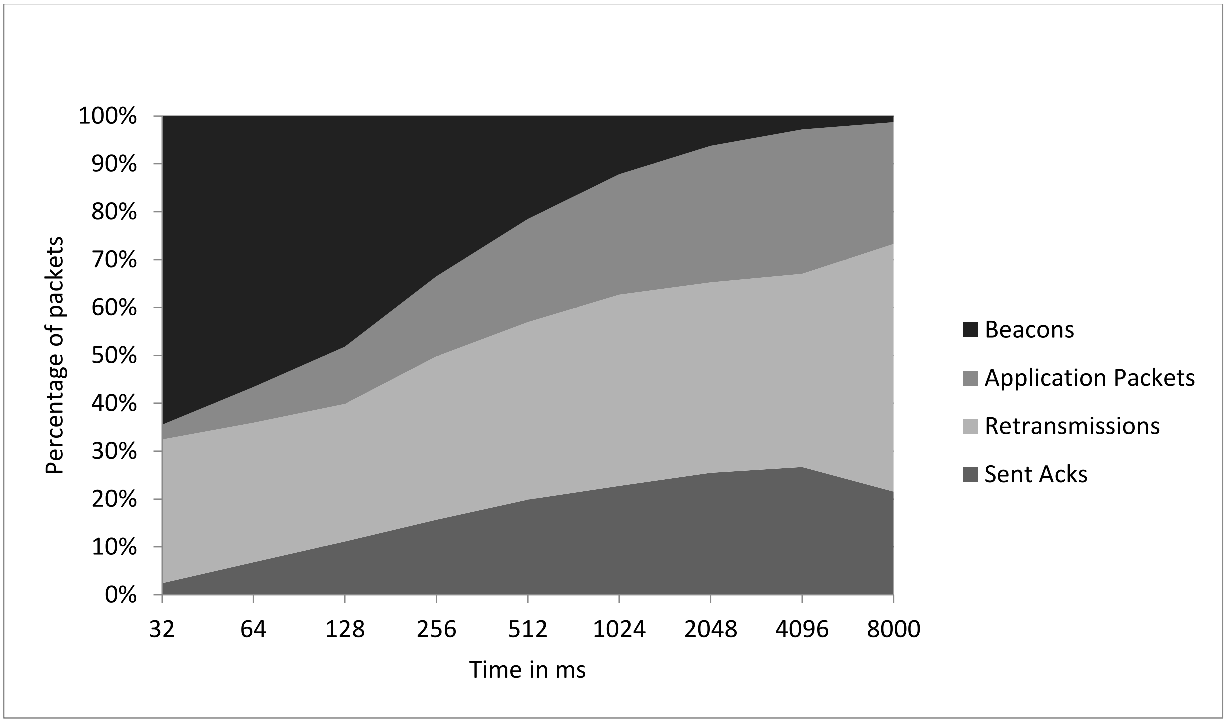


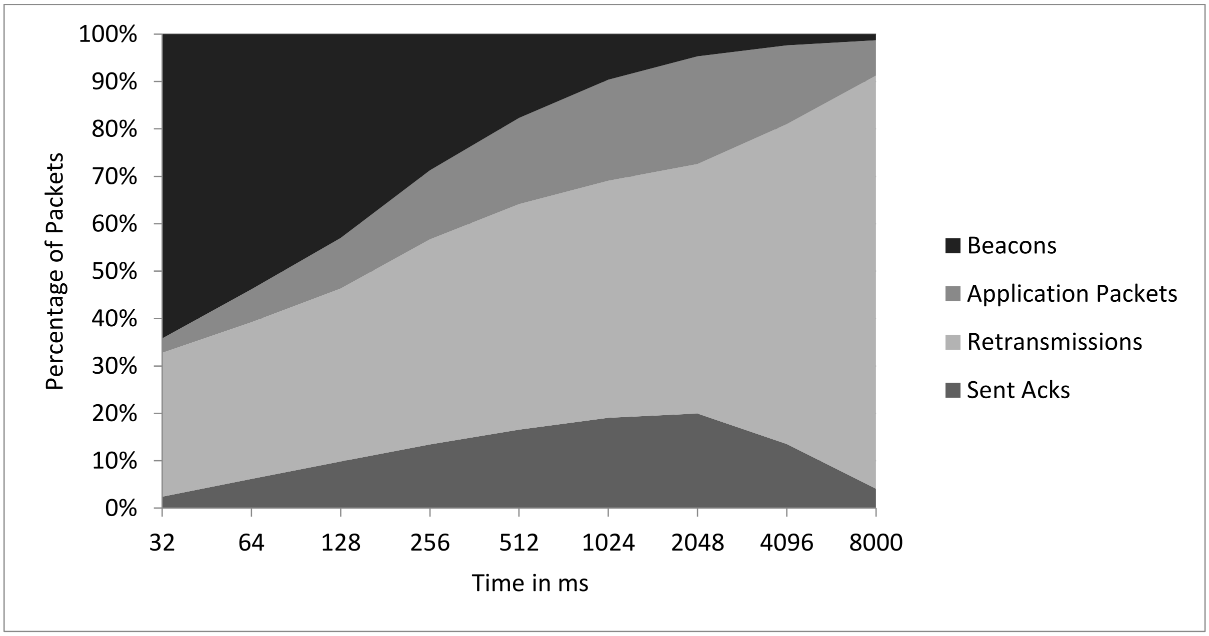
Figure 6.39: The variation in PRR with the variation in beacon interval.

There is an increase in the network's performance when the beacon interval is around 128-265 ms, then the PRR drops. Figures 6.40 to 6.42 will show the breakdown of the transmitted packets in the network. The ratio between application packets, beacons, retransmitted packets and acknowledgments is shown.

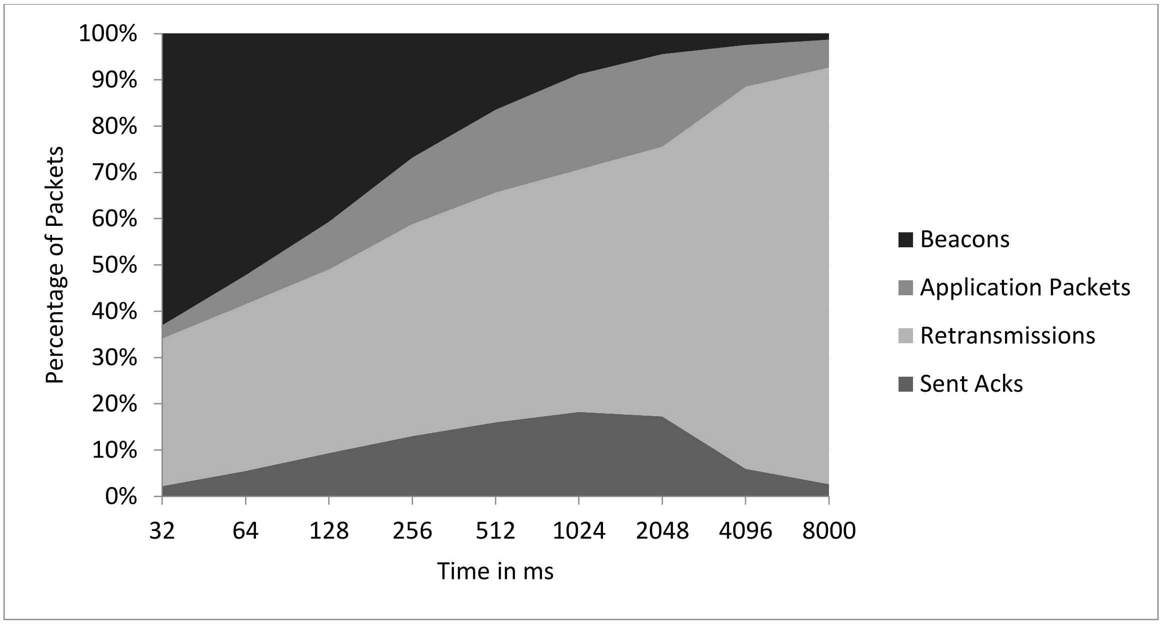
When the beacons are sent out frequently they overwhelm the network, but when the beacon interval increases the retransmissions overwhelm the network.



6.40: Breakdown of packet transmissions in the network with variation in beacon interval for the nodes with 1m/s speed.



6.41: Breakdown of packet transmissions in the network with variation in beacon interval for the nodes with 3m/s speed.



6.42: Breakdown of packet transmissions in the network with variation in beacon interval for the nodes with 5m/s speed.

The results show that choosing a proper beacon interval affects the performance and the overhead in the network. So, it is possible to further increase the performance of the network when choosing different values for the beacon interval.

6.4.2 Packet Origins

In order to understand and investigate FNA-CTP, it is important to know the percentage of packets reaching the sink from mobile nodes, and the percentage of packets reaching the sink from static node. In order to do that, the trace file of the simulation is studied, because it provides step by step documentation of each scenario. End statistic of a simulation can only provide numbers and diagrams, but the detailed second to second analysis will complete the picture. In order to do that, and since trace files are very large and complex, only two scenarios are considered for analysis. Both scenarios are with 90 nodes; 25 and 45 fixed nodes. The mobile node speed was 1m/s. The first scenario had a PRR of 10.9%, the second had a PRR of 92.81%. The values for the PRR are only for the single scenario.

25:90 node network

In this scenario, only 32.4% of the packets received at the sink are from static nodes. The reason for looking into the source is to investigate if the static nodes only act as backups or as main source. In this case, 67.6% of the packets were received from mobile nodes.

It is important as well to understand the number of hops required to reach the sink; this is where the trace file becomes important. Statistics will not provide details about each application packet received at the sink, only the trace file will provide them.

Tracing each packet to the sink, we get the following data:

- The minimum required number of hops to reach the sink was 0, this means the nodes had direct communication to the sink.
- The maximum number of hops was 27.
- 12 packets (3.75% of the received packets) used more than 16 hops to reach the sink.
- The rest of the packets had a maximum of 3 hops to reach the sink.
- Only one of the fixed nodes had direct communication to the sink and hence its packet had 0 hops to the sink.
- The rest of the packets with 0 hops where all mobile nodes that happened to be around the sink and in its coverage range for longer times, due to their location and direction of movement in that scenario.
- Nodes traveling together in parallel and close to each other used one another as parents.
- It is seen that the mobile nodes change their parents while the fixed nodes didn't change their parent on average.

Important to note, these results are only for one scenario; repeated simulation scenarios mean different node starting locations and moving directions.

56:90 node network

In this scenario, the application packets from most of the nodes are received at the sink, which can be driven from the high PRR percentage.

When looking at the number of parent changes in this scenario, it can be seen in the results that the fixed nodes in this scenario chose their parent nodes and didn't change them again; a few fixed nodes changed their parents once or twice. The mobile nodes switched their parents frequently up to a maximum of 17 times.

- The results showed that the number of hops required reaching the sink were in the 4-10 hop range.
- Some packets reached a maximum of 50-65 hops, when looking into their details it showed that they got delivered to the sink with a 20 second delay compared to other packets with the same sequence number.

6.4.4 Concluding Remarks

It is clear that the network's performance would be higher if the number of static nodes is higher than the number of mobile nodes, as the routing protocol was originally designed for static networks. However, the overall number of nodes in the network plays a key role as well, if the network is denser with a higher number of nodes the PRR is higher even if the percentage of static nodes is lower. Therefore, when designing a network with a certain performance in mind, the number of nodes is an important factor if it is a design parameters. In other cases the number of nodes is an input parameters. For example, if the nodes are placed on soccer players, the number of nodes is fixed and cannot be changed.

One of the cost factors to take into consideration in the network is the lost packets due to retransmission, which is the cause of most network overhead. In all the previous scenarios we limited the number of retransmissions to 10 for the mobile nodes and to 20 for the static nodes. The number of retransmission can be varied and fine-tuned according to the application or to the node speed. For example, for higher node speeds the chance of the nodes staying close for longer times is very low, hence there is no need to keep the number of retransmissions to 10, it can be lowered, and this reduces the overhead due to retransmissions.

The control overhead was only a fraction of the retransmission overhead. However, increasing the beacon frequency can increase the PRR, as outlined in section 6.4.3. Similar to the number of retransmission, the beacon interval can be varied according to node speed and application requirements.

The field size is an input parameter for the design, when we increase the size of the network, up to 250 by 250 meters^2, we see an improvement in the PRR when the percentage of static nodes increases. However, it takes a high number of static nodes to increase the performance. The number of hops is another factor than leads to performance degradation. Therefore, there should be a relation between the node degree, the transmission range and the network size in order to achieve the best possible configuration.

It can be argued that the performance of the network is not high if more than 50% of my network is mobile; this is similar to the results in [10]. Looking at the results from another angle, we can say that FNA-CTP performs well in scenarios where the static nodes are interconnected. Based on the previous simulation results, we can state the minimum required number of static nodes in the network to insure a high PRR.

Chapter Seven

Minimum Requirements for FNA-CTP in mobile WSNs

This chapter aims to provide suggestions and minimum criteria to design a mobile WSN network that uses FNA-CTP. First, the optimal design goal is explained in a diagram. Then, using the results of the performance sensitivity analysis, a set of tables is produced, to propose the network requirements to achieve the best performance in the network in terms of PRR.

7.1 Ideal Design Goal

Figure 7.1 shows a simple diagram of how the design of the network is ideally performed. There are specifications for the network that are inputted, based on these inputs the design parameters are chosen to achieve a specific cost.

- Input parameters: which can be the network area *A*. The number of nodes can be an input parameter or a design parameter. The data rate at which the nodes sense and send their data towards the sink. If the node mobility pattern is known, it is considered an input parameter including the speed of the mobile nodes.
- Design cost: It is important to know the goal of the design. It can be a minimum threshold for the PRR based on the application requirement. It can be the overhead cost to reach a given PRR. Time delay in time constrained applications can be a design cost as well, especially when quick delivery to the sink is a requirement.

- Design parameters: based on the input parameters and the required outcome, design parameters are chosen such as the number of the static nodes and the transmission power of the static nodes.

When designing a network usually the field size is known. The number of nodes is specified in some cases, but it can be a design parameter if an optimal number of nodes might be suggested. The node speed plays a role as well in the design.

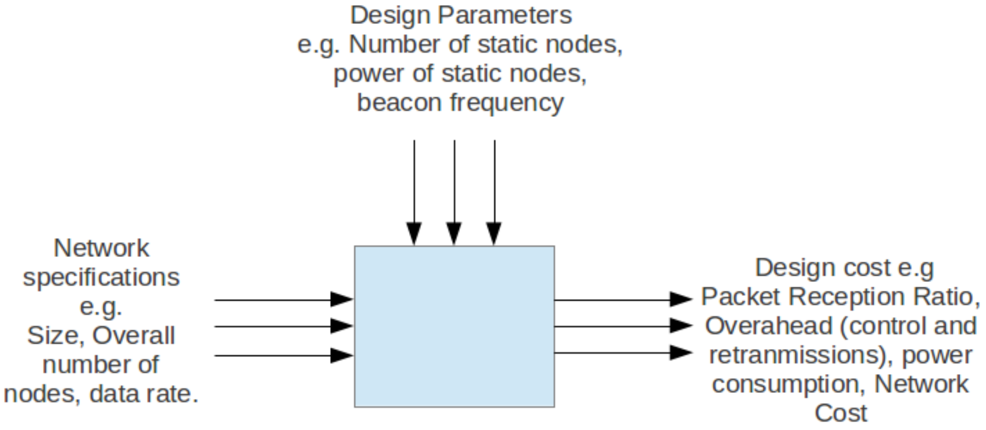


Figure 7.1 Ideal Design Goal.

7.2 Network scenarios and their minimum criteria

The scenarios are divided into two different categories: based on the number of nodes and based on the node speed.

7.2.1 Scenarios based on the number of nodes

Table 7.1 shows the network requirements, in terms of number and transmission range of static nodes, to achieve the best PRR for a specific number of nodes. The table suggests the best number of static nodes, or the maximum number of mobile nodes to achieve the highest PRR. The table shows as well the best scenario for a comparable PRR with less static nodes or less transmission power. Table 7.1

concludes all results of chapter six for large networks with different node speeds. The results shown are all for the highest possible PRR.

Table 7.1: minimum requirements mobile network for a different number of nodes.

	Best scenario			**Best scenario with constrains**		
Number of nodes	**PRR**	**Scenario**	**Transmission power**	**PRR**	**Scenario**	**Transmission power**
45	80%	30:45	High Power	-	-	-
63	86%	36:63	High Power	84.5%	36:63	Medium Power
90	89.9%	56:90	High Power	85% 88.7%	56:90 42:90	Low Power Medium Power
108	89.76%	72:108	Low Power	86.54	64:108	Low Power

If the network requires a PRR that is greater than 70%, then different network scenarios can be chosen as shown in table 7.2. The table doesn't provide the best scenario; it includes the first scenario that exceeded 70% PRR

Table 7.2: possible scenarios to achieve a PRR greater than 70%

Number of Nodes	**PRR>70%**	**Scenario**	**Transmission Power**
45	80%	30:45	High Power
63	74.5%	36:63	Medium Power
90	72.3%	49:90	Low Power
108	75.2	64:108	Low Power

If the network tolerates a PRR of greater than 60% then the scenarios are shown in table 7.3. The first scenario that exceeds the PRR of 60% is included in the table, no constraints are considered.

Tale 7.3: Possible scenarios to achieve a PRR greater than 60%

Number of Nodes	PRR>60%	Scenario	Transmission Power
45	63.%	25:45	High Power
63	61.8%	30:63	High Power
90	72,3%	49:90	Low Lower
108	66.2%	42:108	Medium power

Example

It can be seen that, if the design goal of the network is to achieve a minimum PRR of 60%, then the network can have up to 60% of its nodes mobile if the static nodes have a slightly larger transmission power than the mobile nodes. If the design goal requires a PRR of higher than 70%, then up to 40% of its nodes can be mobile when all nodes have the same transmission power.

7.2.2 Minimum requirements based on node speed

If the number of nodes is a design parameter as well, then table 7.4 provides the best number of overall nodes and the best number and transmission range of fixed nodes to achieve the best performance.

Table 7.4: Minimum requirements for mobile network based on mobile node speed.

	Best Scenario			Best scenario with constraints		
Node Speed	PRR	Scenarios	Transmission power	PRR	Scenarios	Transmission power
Low Speed	89.9%	56:90	High Power	89.8%	72:108	Low Power
Medium Speed	81.6%	72:108	Medium Power	80.4%	56:90	Medium Power
High Speed	83%%	72:108	Medium Power	82.4%	56:90	Low Power

If the network can have a minimum of 70% or 60%, then the chosen scenarios can either have the lowest node density or the highest number of mobile nodes. We chose the scenarios with the highest number of mobile nodes to be included in tables 7.5 and 7.6.

Table 7.5: Possible scenarios to achieve a PRR greater than 70%

Number of Nodes	**PRR>70%**	**Scenario**	**Transmission Power**
Low Speed	70.9%	49:108	Low Power
Medium Speed	72.3	49:90	Low Power
High Speed	70.1%	49:108	Medium Power

Table 7.6: Possible scenarios to achieve a PRR greater than 60%

Number of Nodes	**PRR>60%**	**Scenario**	**Transmission Power**
Low Speed	62.3%	42:90	Low Power
Medium Speed	64.9%	42:108	Medium Power
High Speed	66.7%	42:108	Medium Power

Example

If the nodes in the network move with walking speed, and a minimum PRR of 70% is tolerated, then we can choose the network with 49:108 nodes all having equal power. However, if the best scenario is required, then by looking into table 7.4, the network scenarios 56:90 or 72:108 can be chosen.

Chapter Eight

Conclusions and Future Works

In this thesis, a performance sensitivity analysis was performed on mixed mobile-static sensor networks. First, to understand and analyze the effect of node mobility in a CTP based network. Second, to investigate the performance of FNA-CTP in different network scenarios by varying different network parameters, in order to provide a framework for designing FNA-CTP networks.

In mixed mobile-static sensor nodes, the number of mobile nodes in the network affects the performance [10]. The increase in the percentage of mobile nodes, in larger networks, leads to a decrease in performance. Looking into network connectedness, when nodes are more than 50% static then the network is considered stationary [35]. However, this doesn't ensure a fully interconnected network. It is still possible to have isolated cluster of nodes that are isolated from the other nodes. When designing a network, it becomes crucial to understand the application specifications, such as the field properties and the mobility pattern of the nodes. It is important as well to know the ideal design goal, such as the required minimum performance or the cost.

The application type, mobility pattern, the speed of the mobile nodes and field properties will set rules and restrictions on the design. In vehicular systems the static nodes are best placed at the side of the roads or at intersections [40]. In the soccer field scenario the sink is best placed at the sides of the field. However, if the mobile nodes are more than one hop away from the sink, having a mix of static and mobile nodes will enhance its performance.

The chosen mobility model in the performance analysis affects the results as well. The random waypoint mobility model is considered one of the most common used mobility models in simulation studies. However, when the destination points are not carefully chosen, the model can lead to an increase in node density in the middle of the field. The destination points are chosen in the field and not often enough at the boarders of the field [41]. The existence of mobile nodes is the main contributor to performance degradation and not the used mobility model [35]. However, the concentration of nodes, in certain areas formed due to poorly chosen destination points in the network, will affect the results as; there is a tendency to choose the destination points in the middle of the field instead of the boarders. Our results showed as well that the speed of the mobile nodes affects the network's performance. When the nodes move with lower speeds, the routing protocol can update itself fast enough to keep up with the link breakages.

In the FNA-CTP network, the ratio between the static and mobile nodes should be carefully chosen; when a network requires mostly mobile nodes, only the minimum number of static nodes should be introduced. Moreover, the minimum transmission power should be used to achieve the best performance. However, if energy consumption and network lifetime is not a cost factor, then higher transmission power and fewer nodes can be considered.

Tables 7.1 to 7.6 only show the best simulation scenarios in regards to PRR, it doesn't take into consideration the overhead in the network. Overhead from control packets and retransmissions can be a cost factor for these chosen scenarios. Section 6.4.2 shows that the beacon interval choice has an effect on the PRR. It is possible to increase the PRR if more frequent beacons are sent, but this causes higher control overhead.

In order to be able to apply the scenarios from Ch.7 for larger sized networks, node degree calculations become important. The goal is to maintain the same overall node degree and the same static to static node degree. The ratio between the node's transmission range and the field size should be included as well in the optimal scenario calculations. Hence, it is suggested to increase the transmission range of the nodes in such a way that, the ratio between the transmission range of the nodes and the field size are close to the ratio in the similar network configuration from Tables 7.1 to 7.6, and maintaining the same FNR.

Future Works

There is a number of possible future works that could be done. First, a location optimization of the static nodes can be conducted. Finding the optimal beacon interval and number of retransmissions based on node speed in order to maximize the performance can be investigated as well. A more detailed node degree analysis in mobile networks is another interesting point of study that can be extended as well on other routing protocols.

Moreover, there has been no power consumption analysis conducted on a mobile CTP based WSN; this can give more insight to the calculation of the network lifetime.

As mentioned in the conclusions, one mobility model might not give enough insight on the network's performance. Hence, it is recommended to investigate different mobility models. It would as well be useful to implement a real implementation of an FNA-CTP network and compare the results to the simulation results.

References

[1] Kumar, P.; Güneş, M.; Mushtaq, Q.; Schiller, J., "Optimizing Duty-Cycle for Delay and Energy Bound WSN Applications," in Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on , vol., no., pp.692-697, 20-23 April 2010.

[2] P. Levis, N. Patel, D. Culler and S. Shenker "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks", Proc. 1st USENIX/ACM Symp. NSDI, pp.15 -28 2004.

[3] Anthony H. Dekker and Bernard D. Colbert. 2004. "Network robustness and graph topology." InProceedings of the 27th Australasian conference on Computer science - Volume 26 (ACSC '04), Estivill-Castro (Ed.), Vol. 26. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 359-368.

[4] Perkings, C., E. Belding-Royer and S. Das. "Ad hoc On-Demand Distance Vector (AODV) Routing." IETF RFC 3561 (2003): 1-37.

[5] Johnson, D.B. & Maltz, D. a, 2001. "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks." Computer Science Department, Carnegie Mellon University, Addison-Wesley, p.139-172.

[6] Lambrou, Theofanis P., and Christos G. Panayiotou. "A Survey on Routing Techniques Supporting Mobility in Sensor Networks." 2009 Fifth International Conference on Mobile Ad-hoc and Sensor Networks. IEEE, 2009. 78-85.

[7] Gnawali, Omprakash et al. "Collection tree protocol." Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009). ACM Press, 2009. 1 - 14.

[8] Sharma, Dixit, Ramiro Liscano, and Shahram Shah Heydari. "Enhancing Collection Tree Protocol for Mobile Wireless Sensor Networks." Procedia Computer Science 21 (2013): 416-423.

[9] Gerla, Mario et al. "Dealing with Node Mobility in Ad Hoc Wireless Network."Formal Methods for Mobile Computing 3465 (2005): 69-106.

[10] Jambli, M. N., Zen, K., Lenando, H., & Tully, A. (2011). "Performance evaluation of AODV routing protocol for mobile wireless sensor network." 2011 7th International Conference on Information Technology in Asia, 1–6.

[11] V.A.Amala Shiny and V.Nagarajan "Energy Efficient Routing Protocol for Mobile Wireless Sensor Network "International Journal of Computer Applications (0975 - 8887) Volume 43- No.21, April 2012.

[12] Handy, M.J., Haase, M. & Timmermann, D., 2002. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. 4th International Workshop on Mobile and Wireless Communications Network, p.368-372.

[13] Kumar, G.S., Mv, V.P. & Jacob, K.P., 2008. "Mobility metric based LEACH-Mobile protocol." Proceedings of the 2008 16th International Conference on Advanced Computing and Communications, ADCOM 2008, p.248-253.

[14] Gnawali, Omprakash et al. "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks." ACM Transactions on Sensor Networks 10.1 (2013): 1-49.

[15] Korbi, Ines El et al. "Mobility Enhanced RPL for Wireless Sensor Networks."2012 Third International Conference on the Network of the Future (NOF). IEEE, 2012. 1-8.

[16] Ko, Jeonggil, Stephen Dawson-Haggerty, Omprakash Gnawali, David Culler, and Andreas Terzis. 2011. "Evaluating the Performance of RPL and 6LoWPAN in TinyOS". Ip+SN 2011.

[17]Radoi, Ie, a Shenoy, and Dk Arvind. “Evaluation of Routing Protocols for Internet-Enabled Wireless Sensor Networks.” 2012 8th International Conference on Wireless and Mobile Communications c (2012): 56-61.

[18]Lee, Kevin C. et al. “A Comprehensive evaluation of RPL under mobility.”International Journal of Vehicular Technology 2012 (2012): 1-10.

[19]Nardis, L. De, Domenicali, D., & Benedetto, M. Di. (2010). “Mobility model for Body Area Networks of soccer players.” Network, (September), 65–68.

[20] Garcia, M., Catalá, A., Lloret, J., & Rodrigues, J. J. P. C. (2011). “A wireless sensor network for soccer team monitoring.” 2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS’11.

[21] Dhamdhere, A., Chen, H., Kurusingal, A., Sivaraman, V., & Burdett, A. (2010). “Experiments with wireless sensor networks for real-time athlete monitoring.” Proceedings - Conference on Local Computer Networks, LCN, 938–945.

[22] Reinhardt, A., Morar, O., Santini, S., Zöller, S., Steinmetz, R., Bloom, C., & Sebastian, Z. (2012). “CBFR : Bloom Filter Routing with Gradual Forgetting for Tree-structured Wireless Sensor Networks with Mobile Nodes”, (June), 1–9.

[23] Jevtić, Miloš, Nikola Zogović, and Goran Dimić. “Evaluation of Wireless Sensor Network Simulators.” Proceedings of the 17th Telecommunications Forum TELFOR 2009 Belgrade Serbia (2009): 1303-1306.

[24] Sundani, Harsh, H Li, and V Devabhaktuni. “Wireless Sensor Network Simulators A Survey and Comparisons.” International Journal Of Computer Networks (IJCN) 2.2 (2010): 249-265.

[25] Stetsko, A., Stehlík, M., & Matyas, V. (2011). “Calibrating and comparing simulators for wireless sensor networks”. Proceedings - 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, MASS 2011, 733–738.

[26] Scalable, 2014. QualNet. Scalable networks. Available at: http://web.scalable-networks.com/content/qualnet.

[27] Varga, András, and Rudolf Hornig. “An Overview of the OMNeT++ Simulation Environment.” Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (2008): 60:1--60:10.

[28] A. K¨opke, M. Swigulski, K. Wessel, D. Willkomm, P. K. Haneveld, T. Parker, O. Visser, H. S. Lichte, and S. Valentin, “Simulating Wireless and Mobile Networks in OMNeT++: The MiXiM Vision,” in Proceedings of OMNeT++ 2009, 1st International Workshop on OMNeT++, (Marseille, France), 7 March 2008.

[29] Boulis, “Castalia, a simulator for wireless sens or networks and body area networks,” version 2.0, User’s manual, May 2009 [Online]. Available: http:/ /castalia.npc.nicta.com.au/. Retrieved: 02/04/2010

[30] Ns-2 [Online]. Available: http://www.isi.edu/ns nam/ns/. Retrieved: 02/04/2010

[31] Akyildiz, I.F. et al. “Wireless sensor networks: a survey.” Computer Networks38.4 (2002): 393-422.

[32] Camp, Tracy, Jeff Boleng, and Vanessa Ann Davies. “A Survey of Mobility Models for Ad Hoc Network Research.” Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2.5 (2002): 483-502.

[33] Xue, Feng, and P.R. Kumar. “The Number of Neighbors Needed for Connectivity of Wireless Networks.” Wireless Networks 10.2 (2004): 169-181.B

[34] ettstetter, Christian. “On the Minimum Node Degree and Connectivity of a Wireless Multihop Network.” Network (2002): 80-91. Print.

[35] Santi, P., and D.M. Blough. “The critical transmitting range for connectivity in sparse wireless ad hoc networks.” IEEE Transactions on Mobile Computing 2.1 (2003): 25-39.

[36] TinyOS, an open-source operating system for wireless embedded sensor networks [Online]. Available: http://www.tinyos.net/. Retrieved: 02/04/ 2010.

[37] Colesanti, Ugo, and Sapienza Università. “A Performance Evaluation of the Collection Tree Protocol Based On Its Implementation for the Castalia Wireless Sensor Networks Simulator.” Science (2010): 1-47.

[38] D. Sharma, “Evaluating And Improving Collection Tree Protocol in Mobile Wireless Sensor Networks.” Master of Applied Science Thesis in Electrical and Computer Engineering, University of Ontario Institute of Technology, July 2011.

[39] Sharma, Dixit, Ramiro Liscano, and Shahram Shah Heydari. “Collector Tree Protocol (CTP) performance in Mobile Wireless Sensor Networks.” Canadian Conference on Electrical and Computer Engineering. 2011. 001548-001552.

[40] Yong, Wang, Chen, Xiao, Li Ding, "A static-node assisted adaptive routing protocol in vehicular networks," in Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, Montreal, 2007, pp. 59-68.

[41] Bettstetter, Christian, and Christian Wagner. “The Spatial Node Distribution of the Random Waypoint Mobility Model.” Proc. 1st German Workshop Mobile Ad-Hoc Network. WMAN (2002): 41-58.