A Directional Preference ETX Measure for the Collection Tree Protocol in Mobile Sensor Networks

By

David Krynicki

A Thesis Submitted in Partial Fulfillment

Of the Requirements for the Degree of

Master of Applied Science

In

The Faculty of Engineering and Applied Science

Electrical and Computer Engineering Program

University of Ontario Institute of Technology

January, 2018

Abstract

There has been a growing interest in Wireless Sensor Networks (WSN) that utilizes mobile nodes for various purposes. These mobile wireless sensor networks tend to suffer from constant link breakages mainly caused by connected nodes moving apart, often moving very quickly. These lost connections require WSNs to constantly repair the network connections; this constant maintenance in turn causes power and packet losses and very noisy network conditions. However a performance extending metric can be implemented in order to reduce the frequency and occurrence of lost links between a parent node and its child. As such a directional preference Estimated Transmissions Count (ETX) measure was developed for the Collection Tree Protocol (CTP) in order to create longer lasting links. This thesis describes and measures the performance of this directional preference ETX measure utilizing various metrics such as Packet Reception Ratio, average number of beacon transmissions per node, Parent changes and various others. The Packet Reception Ratio metric is primarily used to compare this directional preference ETX measure to other popular WSN algorithms such as M-Leach, Geographic Greedy Forwarding and as well regular CTP due to the differences in topology between these algorithms. Based on the packet reception ratio the directional preference ETX measure improves the performance of CTP such that it is capable of outperforming M-Leach in various scenarios.

Acknowledgements

I would like to deeply thank my supervisor Dr. Ramiro Liscano for his guidance, patience, and support throughout this work. I have learnt much from you, not only throughout the course of my thesis work but also throughout my undergraduate degree. From the bottom of my heart thank you.

Thank you also to my family, my parents Jadwiga and Janusz and brother Alex for all their support and help over the past few years.

Contents

List of Figures

List of Equations

List of Tables

List of Abbreviations

CTP	Collection Tree Protocol
DP-CTP	Directional Preference Collection Tree Protocol
ETA	Estimated Time of Arrival
ETX	Expected Transmissions Count
GGF	Greedy Geographic Forwarding
GPS	Global Positioning System/Satillite
LEACH	Low-Energy Adaptive Clustering Hierarchy
LOGR	Joint Localication and Geographic Routing
MANET	Mobile Ad-Hoc Network
METD	Minimum Estimated Time of Delivery
PRR	Packet Reception Rate
THL	Time Has Lived
VANET	Vehicular Ad-Hoc Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Wireless Sensor Networks

Wireless Sensor Networks have been developed and researched for many years, and have been improved upon so that they are now capable of monitoring large swaths of land and sea and have even been utilized within buildings. There is a multitude of routing solutions available to optimize multi-hop WSNs to perform such that they are either energy efficient, decrease overhead or decrease amount of dropped packets all of which can be classified into three categories: Data Centric, Hierarchical Cluster based and Location based routing as has been mentioned by Sharma [1]. Some of these methods include Collection Tree Protocols (CTP), Geo-Routing, Mobile Ad-Hoc Networks (MANET), Clustering Algorithms. However as technology has improved the ability to make nodes within a network move under their own power to survey areas hazardous to humans or to place nodes in environments that make them move such as flowing rivers or further still attach nodes to humans or animals to measure vitals have provided routing techniques with challenges in routing and forwarding of data that previous stationary node routing techniques have difficulty with. These new mobile wireless sensor networks now have to be able to handle situations where combinations of some or all sinks and or nodes within a network are mobile. Further still situations where the mobility is beyond control of the network such as when nodes are attached to animals and their paths are impossible to predict.

A Wireless Sensor Network (WSN) is a collection of devices commonly referred to as sensor nodes. The purpose of these nodes within a network can vary, their primary task can include the collection of a variety of data utilizing various sensors within a single node to only measuring a single variable per node in order to pass on the data immediately or collect it for future use. Since each of these nodes are constrained to a limited amount of power, processing and data storage, their secondary function; the forwarding of data towards a sink; is extremely important in order to utilize the collected data of each node.

Generally a node within a network is usually made up of a core set of parts such as a sensor in order to collect data, external memory in order to store collected data, a transceiver in order to communicate with other nodes, a processor or microcontroller to process sensed data and to use the transceiver to send and receive data from other nodes, and finally a power source to power everything within the node.

A typical WSN will contain one or more sinks which are devices that collect the data that the nodes within the network send to be handled. Within these networks the nodes are structured in one of two general ways. The first is a network of nodes that are a single hop away from the sink meaning that each node in the network is directly connected to their sink. The other structure is a collection of nodes that require multiple hops for a data packet to reach the sink. This type of network utilizes nodes between the sink and the furthest nodes as relay points to forward the data to the sink. Both methods have their pros and cons. Single hop networks have better reception and power consumption but have very limited coverage or range. Multi hop networks in contrast can cover large areas however suffer from wireless communication collisions and must use extra power to forward data.

1.2 Challenges of WSNs

A Wireless Sensor Network as a whole is the collection of sensor nodes along with a sink. Usually these sensors can be deployed in an ad-hoc manner and are generally without any infrastructure, apart from the sink which is a key element within the network. However without sensor nodes there is no network and yet these nodes are the weakest part of the network as they are the main source of constrains within the network. The lifespan of a WSN tend to be limited by the amount of energy they store and the amount of power drain caused by communication, sensing, and data processing [2]. Some methods of prolonging the lifespan include aggregation of data which reduces the amount of power consumed by the radio[3], [4] or even reducing the number of times update packets are required [5].

Another challenge that is faced by WSNs is the robustness of the network although networks are capable of recovering from lost links such as various fault tolerances [6] that have been implemented in different WSNs. Topology of networks are also important in the performance of and maintainability of a network , if there is low or limited connectivity between nodes a network cannot be sustained [7]. Link longevity is another issue WSNs must deal with, link losses between nodes mean that the network will need to expend additional resources in order to repair lost links.

1.3 Problem Statement

Mobility introduces an increase to overhead to routing protocols within WSNs due to the fact that nodes that are in communication may leave each other's radio range causing the protocol to have to find a new parent. This causes the network's topology to be fluid which in turn forces the routing protocol to rebuild the routing tree. In order to for a routing protocol to cope with such fluid changes in topology it must quickly react to lost links, repair only the portions that are broken, be able to identify loops and choose a parent that would be unlikely to be lost.

1.4 Thesis Statement

The goal of this research is to document the design and implement of an improvement for a CTP based network utilizing mobile nodes to minimize the need to rebuild links between nodes improving connectivity and reducing the amount of overhead caused by beaconing. This work will describe two modifications, the first is the adjusting the method in which child nodes select their parents in a network. As well as creating a variable beaconing delay which is dependent on the direction of travel that a parent and child are headed in, such that when both vectors of travel are compared the nodes will seem to be relatively static to each other or otherwise moving apart at a very slow rate.

1.5 Methodology

The methodology to complete this work began with a literature review of related works and backgrounds, in order to create a foundation of understanding of the structure and performance of various algorithms that handle mobile WSNs utilizing different topologies. Once a better understanding of these networks was attained a hypothesis of a simple modification to CTP in order to improve connectivity was made and applied. This modification was then tested by simple comparison against an unmodified version of CTP under different conditions including enabled and disabled trickle algorithm which is incorporated in CTP as well as varying speeds, node count and network area. Performance tests on the modification were conducted in order to determine how the algorithm performed under different stressors such as: large areas, different speeds, small to large number of nodes. Finally a comparison against different types of popular algorithms utilizing different topologies to determine how the modification of CTP measures up.

1.6 Contributions

Previous research has demonstrated that mobility poses a challenge in WSNs [1][8]. As such the main contributions of this work are:

- A directional preference ETX measure for CTP which improves connection longevity between a parent and a child traveling in approximately the same direction
- Criteria based trickle algorithm enabling metric which simply enables the trickle algorithm if a parent and child are traveling in the same direction.

Additionally the preliminary results found in this thesis were published in the following venue:

- Krynicki D, Liscano R. (2017). Directional Preference Collector Tree Protocol for Mobile Wireless Sensing. Lecture Notes on Data Engineering and Communications Technologies, vol 7. Advances in Network-based Information Systems (NBIS 2017), Toronto, Canada (339-350) [9].

1.7 Thesis Structure

This thesis is organized into six chapters. The first chapter contains the introduction which covers a brief on Wireless Sensor Networks, the challenges that they must deal with, a thesis statement, methodology and contributions made. The second chapter begins with an overview of the related work that is associated to the research conducted. The following chapter describes the Collection Tree Protocol as well as the design modification made to it. The fourth chapter describes the network model that was used for simulating and collecting data through this work. The results of all the various simulations are found in the fifth chapter, which contains data on the initial comparison between modified and unmodified CTP, performance analysis of the modified CTP from here on named Directional Preference CTP (DP-CTP) and finally a comparison against other popular routing protocols. Finally the sixth chapter contains the conclusion and future work.

Chapter 2

Related Work

Mobile nodes within Wireless Sensor Networks provide additional challenges to any network. The biggest challenge faced is the routing of data, mobility within a system such as this can cause path loss as well as forwarding issues which in turn may create collisions and congestion while trying to retransmit missed packets. There can be three types of wireless sensor networks based on the type of nodes used: protocols where all nodes are static and sink is mobile, protocols where nodes are mobile and the sink is static and finally where all nodes and sink are mobile. Additionally different types of protocol solutions have been designed and created to support these types of networks.

2.1 Routing Topologies

There are three common types of network topologies that are utilized in wireless sensor networks, these being Tree, Clustering, Chain, and Flooding topologies. This related work section will only cover the first three types.

2.1.1 Clustering Topology

These types of topology protocols utilize cluster heads to service a group of nearby sensor nodes which collect and aggregate data. Once data is collected and prepared it is then forwarded to the sink. These types of protocols however efficient in forwarding data suffer from lots of overhead from cluster membership management [8]. There are a variety of clustering protocols with different methods used to select a cluster head.

LEACH a popular Clustering algorithm documented by Heinzelman et al. [10] randomly selects a cluster head to distribute the energy load amongst all the nodes within the network. Conventional clustering algorithms at the time used static pre-selected cluster heads that would be utilized for the life of the network [10]. LEACH is broken up into rounds; at the beginning of a round the set up phase occurs where clusters are created by nodes electing themselves to become cluster heads and broadcasting their availability to their neighbours who which in turn decide to choose to which neighbouring cluster head they shall belong to. The next phase is the steady state phase where data transmission occurs between these nodes, specifically from a node to a cluster head and the data aggregated before being forwarded to the sink. This second phase is the longest phase. Once a round is over the cluster heads are reselected by way of randomization, this is done to prevent a single node from depleting its power source by receiving and processing its clusters packets before forwarding.

One such protocol similarly to CTP was modified to be better utilized in mobile scenarios. Kumar et al. [11] improved upon LEACH and created a Mobility Metric Based LEACH mobile protocol, which is similar to LEACH in that it selects cluster heads which service local nodes and aggregate data before forwarding it towards the sink. This protocol operates in two phases: set-up phase and steady-state phase. The first phase or set-up phase is the when all initial cluster formation occurs while the steady-state phase handles the data transfer between nodes. Normally during the set-up phase a node is selected as the cluster-head based on its power resources however in Mobile LEACH the relative motion of the cluster head to its possible children is taken into account. Since cluster heads handle lots of traffic they are prone to failure, as such LEACH would normally randomize the selection of the next cluster head. When mobility is an issue this would cause problems because a randomly selected node might move out of reach faster from the surrounding neighbour nodes. Mobile LEACH handles this by utilizing a mobility factor in the selection process to choose a node that does not leave its cluster area which means it would be a more stable cluster head.

Oh et al. [12] propose another version of clustering; called Dynamic Direction Vector Hop clustering algorithm, this algorithm takes into account the mobile nature of the nodes such as velocity and direction in order to form clusters. The idea is to gather nodes into a cluster that are traveling in a similar direction and speed so that the frequency of rebuilding the clusters is lowered. First the protocol elects the cluster head using coverage. This is accomplished by the base station by counting the amount of nodes and their direction within each region. Within each region, the base station then selects the most common direction that the nodes are traveling and then elects a node traveling in that direction to be a cluster head. Once a cluster head is elected it broadcasts its designation and nearby nodes will choose it as a parent only if the direction and velocities are similar to each other.

The ideas used in these clustering protocols for cluster head selection can also be applied to CTP's selection methods for a single parent node. Although selecting and maintaining a cluster head carries with it more overhead and communication amongst more nodes then does a tree protocol where maintenance is only required between a parent and a child node.

2.1.2 Tree Topology

The other Topology style utilizes protocols that are tree based which are designed to create connections that branch from the sink node towards the rest of the sensor nodes which create in essence a network map that would resemble a tree.

One of the more popular tree based protocols namely CTP is a tree-based collection protocol that manages dynamic paths to sinks from sensor nodes. This protocol is also responsible for handling data transmission along these paths. Additionally it is also capable of maintaining a network with multiple sinks, which is possible because CTP is address free and sensor nodes are not responsible for selecting a full path to the sink, but only towards a parent which in turn selects its own parent; this is repeated until the sink is selected as a parent [13]. In other words the sink is chosen implicitly by choosing a neighbour to send data to [14]. Parents are chosen based on beacons that are broadcast which are used to assist in generating paths between nodes in order to create a path to a sink in network [15]. Tree creation and maintenance for CTP is further described in section 3

Overview of CTP.

Otman [8] investigated slight modification to CTP to attempt to improve its Packet Reception Ratio by incorporating support nodes that are static. These support nodes were used such that if a regular sensor node would lose its parent it would then automatically connect to a support node before finding another parent. These support nodes were also given a higher buffer size to be able to handle large volume of disconnected nodes. Additionally it was found that additional modifications such as removing the trickle algorithm and lowering the amount of retransmissions provided better results.

Hassanzadeh [16] found that the original reaction to timeouts cause massive changes to the tree structure, especially when a node close to the sink loses its parent. When this occurs all the child nodes will disconnect and then look for new parents which would cause a lot of overhead for the rebuilding of the tree. His solution was to change the ETX to the maximum value, which would only cause the next child to find a new parent thereby preserving the branch. After a few beacon broadcasts the nodes on the branch would automatically update their ETX over time as they received their new parent ETXs. This gradual change does have some growing pains during the tree update but is much less destructive and energy consuming than the original.

Flathagen et al. [17] developed a hybrid version CTP named O-CTP which is an opportunistic variant of CTP. It still utilizes the original routing portion of CTP but utilizes a new opportunistic routing method, which when certain conditions occur will trigger the switch between the two different methods of routing. This opportunistic method takes advantage of the broadcasting nature of this type of network and assumes that all neighbours in the direction of the sink are listening and one of the nodes is able to forward the data. This portion of the algorithm does however assume that the neighbour nodes are capable of hearing the packet, broadcasting the packet to a subset of neighbours lowers the probability of lost packets.

2.1.3 Chain Topology

Chain topology can be observed mostly within Geo Routing protocols as the nodes generally connect to each other basically by finding the furthest node that is still in range possible or some similar criteria. This method in essence creates chains where nodes that are close to each other have a good chance of never sharing a common parent even though they are very close to each other. It is still possible for some chains to appear as branches normally found in tree based protocols however all nodes that are within range to the same parent do not necessarily choose the same parent. This is unlike tree protocols where a parent is an optimum hop node due to a particular metric even though it is not an optimum distance point between node and sink.

Leontiadis et al. present GeOpps [18] which is a geo routing protocol that exploits the available information provided by navigation systems to find the best location to forward data. It accomplishes this by calculating a nearest point on its suggested route that is closest to the destination of the packet. Additionally it makes a decision based on its neighbours and if they will reach the nearest point sooner or their path will be closer to the destination. This is simply calculated by calculating what is known as a Minimum Estimated Time Of Delivery for the packet (METD) the equation used by GeOpps [18] METD = ETA to Nearest Point + ETA from Nearest Point to Destination.

Zhu et al. [19] utilize a probabilistic approach utilizing directions and relative angles to the sink node to determine the optimal time to broadcast a packet specifically if the node will be a single hop from the sink. This is achieved utilizing three rules, the first rules indicate that if the node is already within broadcasting distance it should transmit its packets. The second rule calculates if the node will be within broadcasting distance of the sink and if it is it will hold its packet until it reaches the optimal location, at that point the node will transmit its packets. The first two rules assume that the Probability is equal to 1 because the node is 1 hop away from the sink. The third rule calculates the probability of the next destination path, if the next destination and its path will coincide with the broadcasting range of the sink to maintain a 1 hop transmission it will hold the packet, otherwise the probability is calculated by incorporating the communication angle and the final decision on whether to transmit or not is based on the data transmission algorithm.

Nasipuri et al. [20] developed a method for nodes within a sensor network to determine their angular bearings based on received beacons. This is accomplished by three fixed nodes that broadcast special beacons which sensing nodes use by comparing the times in which it receives a beacon from the same static node and uses that to determine if it is headed towards or away from it. Each directional broadcasting node sends its beacons at regular intervals. The times can be used to extrapolate displacement and then using trigonometry calculate the angular direction of travel for the node. The results in [20] show that the localization scheme performs with high accuracy whilst not being very complex. They indicate that the error due to the beam width of the static nodes falls within 15 degrees.

Seada et al. [21] describe a handful of forwarding strategies for lossy networks all of which are based on distance based forwarding, to improve parent selection. Original Greedy forwarding forwards data to a node that is within range but is closest to the sink. Distance based Blacklisting prevents nodes from blindly selecting nodes but gets rid of options based on if the node resides in an area that is near the furthest reach of the radio, this is an effort to ignore nodes that may have low reception rates because of how far away they are. Absolute reception based blacklisting eliminates nodes that have a reception rate that is lower than some criteria amongst the nodes that are the furthest away but still in range. Relative Reception based blacklisting black lists all nodes that have a reception rate below some threshold and then the node furthest away but still in range is then selected. Best reception neighbour is a method that only selects a node based on the best packet reception ratio. Best packet reception ration and distance utilizes the product of the reception rate and the distance improvement that would be achieved by selecting a particular node, the distance improvement is calculated utilizing the following formula:

$$1-\frac{distance\ of\ neighbour\ to\ desitnation}{distance\ of\ node\ to\ destination} \tag{1}$$

They found that reception based forwarding strategies perform better compared to distance based strategies. Additionally they found that joining the reception strategy and the distance measure performed the best in their simulations.

Bertanha et al. [22] proposes a Greedy Geographic Forwarding (GGF) based routing protocol LOGR (Joint Localization and Geographic Routing-based Data Dissemination) that uses packets broadcast to determine locations. In a basic GGF protocol a node selects a parent that is within range but is as far away from the node as possible. Since small and cheap nodes would generally not contain a GPS, LOGR utilizes the packets that are broadcast by neighbours and the sink to determine a nodes location, this is accomplished by a trilateration algorithm. Additionally nodes that determine their location based on the sink are more reliable because the sink is fitted with a GPS while regular sensing nodes are not. LOGR utilizes static sensing nodes and a mobile sink, in order to create paths to the sink the sink must broadcast its location, each time it broadcasts this it applies a time to live value that is reduced each time a sensing node forwards this information. LOGR uses an algorithm that incorporates the distance to the sink, the neighbouring node, the power of the neighbour node and the weighting factors for the distance and power variables in order to determine which neighbour is the best node to forward to. The paper [22] goes on to indicate that the algorithm performs equally well as GGF however it has been able to improve the network lifespan.

2.2 Simulators

With the aim of investigating a protocol some form of testing is required; the reliable method would be to construct a real network and apply the protocol in question and study the results. This however is an extremely expensive process both in time and money. A more appropriate approach is to analyze the protocol within a simulator. This approach allows for many different scenarios of varying lengths of time to be conducted quickly, automatically and with very low costs. A few surveys available have covered the functionality, pros and cons of various simulators [23], [24] such as ns-2, OMNet++ and Castalia, TOSSIM, COOJA/MSPSim just to name a few. Furthermore additional surveys and papers cover the reliability of results provided by various simulators [25]. As such Castalia [26] allows for the user to easily utilize and modify any protocol, it also contains accurate radio and channel models. This simulator is also capable of supporting multiple types of nodes within a simulation as well as multiple sinks. Most importantly it is able to simulate a mobile network with simple and complex node paths as well as supporting custom mobility patterns. As such OMNet++ and Castalia were utilized to complete the simulations for this work.

2.3 Mobility Patterns

Mobility patterns themselves although innocuous at first glance are extremely important in the simulating process. Depending on the simulation, the ability to mimicking traffic for Vehicular Area Networks VANETs is key in developing and testing a suitable protocol or the ability to generate node movement to simulate foot traffic in an area to forward data through a crowd of personal cell phones. As such a research paper [27] has investigated the impact of mobility patterns on the efficiency of data forwarding in MANETs as well as survey by Camp et al [28] review various mobility models. Additionally Camp et al [28] concluded that a network protocol can provide greatly varying results based on which mobility model is selected, however the same mobility model can also provide greatly varying results based on the parameters it is given. Additionally Lagkas et al [27] concluded in their research paper that “In general, the more independent, dynamic, and less coherent the node movement is, the less paths can be created, hence, less transmissions can be successfully completed” [27]. This statement backs up why the random way point mobility model is so popular; a mobility model that puts a high strain on a network provides a worst case scenario for the network and is why this mobility model was selected.

2.4 Connectivity

Additionally Connectivity is a concern as it is extremely important that in a network without a good level of connectivity, the ability to communicate with all the nodes in a network is impaired. Various studies have been conducted about the connectivity of a network be they mobile or static networks. A successful network should ideally have all nodes connected to a sink and where no nodes or clusters are isolated, meaning that the minimum node degree should be equal to 1 which would ensure no node is isolated [29]. However because of faults or node failures or even high mobility a node which only has one neighbour could easily become isolated. Xue et al. [29] concluded in their paper that the following equation:

$$Minimum\ average\ node\ degree = 5.1774 * \log n \tag{2}$$

Indicated how many neighbours are required to keep connectivity healthy, where n is the number of nodes in a network.

Additionally transmission range is also hugely important to the connectivity of a network as such Santi et al. [30] investigated the correlation between the area of a network, the number of nodes, and the transmission range and how they would affect a two dimensional network. They found that the following equation for a two dimensional network:

$$R^2 n = l^2 \log_2 l \tag{3}$$

Where R is the radio range, n is the number of nodes and l is the length and width of the network area which can be used to determine the required criteria for a connected network. However this equation for mobile situations should be considered as a lower bound as mobility requires additional neighbours or larger radio ranges to stay connected more reliably.

Chapter 3

The Proposed Directional Preference Collection Tree Protocol

This chapter provides an overview of the protocol named the Collection Tree Protocol (CTP) which is the basis of Directional Preference-Collection Tree Protocol (DP-CTP) and how the network is constructed and maintained. Details on the network tree creation, maintenance, and the original ETX calculation. This chapter also covers the rational for the modification of the ETX measure for mobile networks as well as the specific changes made to create DP-CTP.

3.1 Overview of CTP

This overview explains the structure and function of CTP in order to more easily explain the change made to create DP-CTP

3.1.1 Tree Creation

CTP's advantage over other protocols is the lack of dependency of an address structure essentially only requiring the location of a parent node. For CTP to operate without the use of addresses it creates a singular tree or multiple trees starting from one or more sinks respectively. This method allows for CTP to manage a direct path to a sink even if a connection is severed.

The creation of the tree is initiated by the sink when it broadcasts a beacon providing all its neighbours with an ETX of 0 (see Section 3.1.5 for how ETX is calculated); meaning that the sink is the final destination and should immediately be selected as a parent by all neighbours that receive the beacon. In situations where multiple sinks are available the beacons contain a tag that identifies which sink or tree the beacon is advertising. Whenever a node receives a beacon it will put the source neighbour into its neighbour table if there is room. If room is not available then it will remove a neighbour from the table if it satisfied one of the following cases:

- A neighbour node has not been updated for extended predefined period of time. This timeout period can be changed depending on the requirements for the application of the WSN.

- A neighbour that does not have a Link quality yet (Q_u or Qb see section 3.1.5). This indicates that the node is not mature and can be determined by the beacons that are received.
- If no nodes in the neighbour table fall into one of the two previous cases then a node with an ETX value higher than a pre-defined value can be evicted. This pre-defined value can be selected based on application requirements.
- If none of the rules above cause an eviction and if the new beacon contains an ETX lower than any in the table currently it will replace the neighbour with the highest ETX. The sink always has the lowest ETX and will always replace a node in this case.

If none of the cases are satisfied the node is then ignored.

Once a node receives a beacon from the sink it will select it as its parent. That node will then broadcast its own beacon advertising its own availability to its neighbours. This new beacon will provide all nearby nodes the ETX of that node which indicates the distance or cost to the sink. Following the same rules nearby nodes will receive the beacon make their decisions based on all received beacons, and repeat. This will occur repeatedly until all nodes are connected to a sink. When a node is selecting a parent from its own neighbour table it will select a neighbour with the smallest ETX. If there are two or more nodes that have the same ETX in the neighbour table which are also the lowest, the inequality sign used in the 'IF Statement' to differentiate ETX will determine if the first node or the last node is selected. If the inequality sign is looking for less than or equal then the last node in the table will be selected otherwise if it is just comparing using less than, the first node in the neighbour table would be selected.

3.1.2 Tree Maintenance

CTP allows for the recovery of connections to the sink in an efficient manner. When a node that supports multiple children becomes disconnected from its parent, instead of instructing its children to find a new path, the disconnected node will search for a new parent to the sink. This method maintains the tree after it was disconnected with only one node requiring a reconnection rather than all children in that branch. Once reconnected overtime the children will receive updated beacons from their respective parents and the system will then contain the correct ETXs. After calculating the new ETXs the nodes will if needed adjust their paths to the sink.

3.1.3 Scalability

CTP's tree formation and maintenance is so simple while utilizing no addresses and relying on beacons to advertise paths allow it to be wildly scalable. This is possible because in an already existing network of nodes adding a new node means that the new node will listen for beacons and simply select the best neighbour in the area and set it as its new parent, effectively attaching itself to a branch of the tree. Assuming that the nodes have infinite power and infinite bandwidth the network could support the addition of additional node indefinitely. Under normal circumstances there is a saturation point, where by the addition of new nodes in a confined area would cause for congestion on all nodes. Replacing dead nodes in a network by introducing a new node by simply placing it in the area can be done indefinitely after a dead node is removed.

3.1.4 Loop Management

If a disconnection occurs within the network and a node must now select a new node from its neighbour list, there is a possibility that a child node might be selected, this includes any level of separation of child nodes. When this occurs a node will recognize that the multi-hop ETX value received is lower than its own, this indicates that a loop has occurred because a child's ETX cost must be higher than the parent because the ETX is the sum of all prior ETXs plus the source node. If this occurs the nodes will request the neighbour nodes to reset the beacon intervals, which will allow for a quicker repair of the loop.

3.1.5 ETX Calculation

ETX values that are used in the creation of the neighbour table are calculated by the Link Estimator. A neighbour cannot be selected as a parent until the 1-Hop ETX value is calculated. The calculation for the 1-Hop ETX value is completed based on the incoming and outgoing link quality. The link quality is calculated based on the number of successfully transmitted data packets sent to a node's parent over a period of time w_u or w_b for outgoing and incoming qualities respectively. This is accomplished through beacons that are sent, when each beacon is sent its sequence number is provided as well as the amount of beacons that were sent by that node, and an incrementing value is kept for each beacon received by that node. The quality of the outgoing link is then calculated by dividing the amount of received beacons by the total number of beacons sent [31] , which is shown by the following equation where Q_u is the quality of the outgoing link, n_u is amount of unicast application packets sent, and n_a is the total amount of acknowledgements from the parent.

$$Q_u = n_u/n_a \tag{4}$$

Similarly the incoming link quality is calculated based on the number of unicast application packets including retransmissions divided by the amount of acknowledgements from the parent, which is shown by the following equation where Q_b is the quality of the incoming link, n_b is amount of received beacons from the parent, and N_b is the total amount of beacons sent by that same parent.

$$Q_b = n_b/N_b \tag{5}$$

With these results the ETX can be calculated using the following equation:

$$ETX_{1Hop} = \alpha_{ETX}Q + (1 - \alpha_{ETX})ETX_{old\ 1Hop} \tag{6}$$

Where the most recently calculated Q_u or Q_b can be used for Q and α_{ETX} has a default value of 0.9.

Once the one hop ETX is calculated, it is added to the ETX of the parent to create the ETX the node will later broad cast.

$$ETX_{node} = ETX_{parent} + ETX_{1Hop} \tag{7}$$

3.2 Rational for the Directional Preference Collection Tree Protocol

The main purpose of DP-CTP is the selection of a parent node that is traveling in the same direction as the child is to reduce the frequency these connections have to be changed. When a connection is stable the tree requires fewer instances of rebuilding which means it can save processing and power, which in turn would allow the network to forward more data instead of focusing on rebuilding failed connections. The aforementioned Trickle Algorithm under these circumstances would in essence be performing in a static environment which means it would be able to help lower the frequency of beacons being sent.

A best case scenario as shown in Figure 1 where Node C has two possible parents, the first potential parent Node B which is traveling in the opposite direction of Node C, and the second potential parent moving in the same direction as Node C but which has a slightly worse ETX. Under regular CTP Node C would connect to Node B and when the range between the two would be too great would have to reconnect. Utilizing DP-CTP parent selection process Node A would be selected, regardless of the slightly poorer ETX, in this case Node C can hold onto their parent for a longer period of time which would reduce the amount of beacons required to maintain the network.

The following examples assume that Node C is unable to connect to the sink.

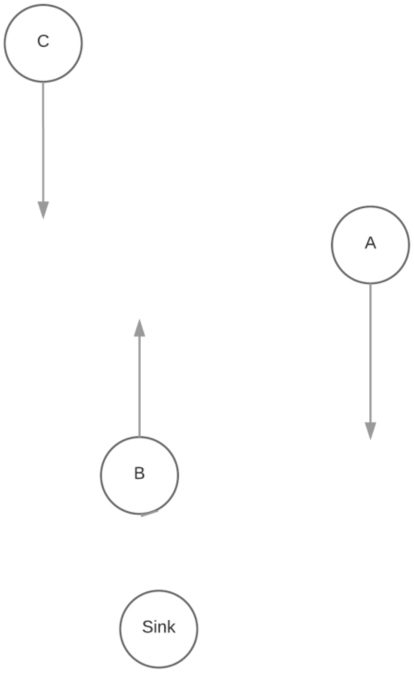

Figure 1: Best case Scenario

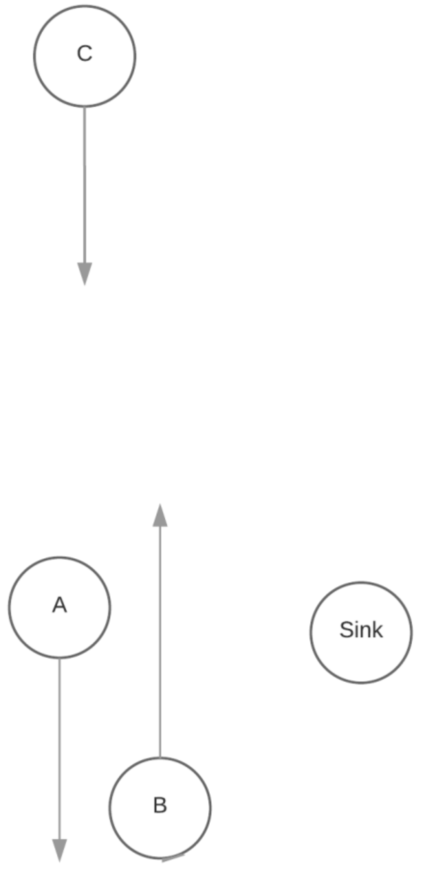

Figure 2: Worst Case Scenario

A worst case scenario shown in Figure 2 shows a case where parent Node A was selected utilizing the DP-CTPs selection criteria for child Node C. In this case Node A will shortly leave the range of the sink and Node C would have to select a new parent, as such the ideal choice in this scenario would have been Node B as it is headed towards the sink, however it was not chosen because the θ calculated would have disqualified it as a parent. As such this selection would mean that the child would have to select a new parent (or the sink). However Node A after it had left the range of the sink would then select Node C as its parent and would then require fewer reconnections as it would stick with its parent longer. This can be demonstrated in Figure 3 where two chains of nodes are moving in opposite directions. Node C in this case would still select the parent moving away from the sink (Node B) however Node B afterwards would select B1, and then B1 would select B2 as their new parents when their previous parents have left the rang of the sink. This selection process would allow for the nodes to maintain a stronger connection to the sink unlike if any of the B nodes would have selected an A node as a parent. Generally speaking the worst case scenario for this protocol when all nodes are taken into consideration is random waypoint model; this is due to nodes having low chances of having the same or similar direction of travel. The ideal mobility model for the network as a whole would be something akin to a two way highway as nodes would only need to reconnect to a new parent once they have left the range of the sink and would need to select a new parent.

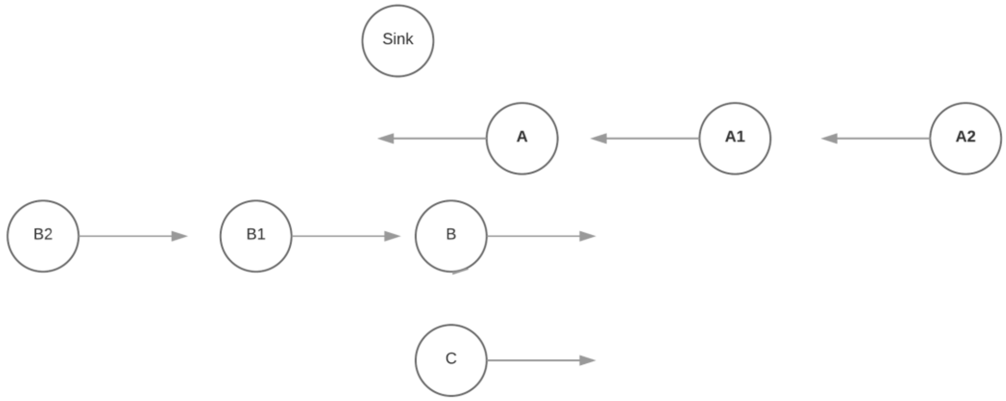

Figure 3: Grouping

3.3 DP-CTP

In an effort to improve the performance of CTP in mobile situations modifications to CTP were necessary. Since mobile nodes tend to lose connections to their parent and neighbour nodes mostly due to leaving radio range and less so to causes commonly found in static WSN, it became evident that an improvement to tree construction or parent selection may provide a better results.

3.3.1 Directional Preference ETX

In an effort to reduce the need to rebuild the tree constantly due to nodes moving away from each other in the network requires an adjustment to the selection of parents. The selection of parents in vanilla CTP is based on just an ETX, if the nodes know their current direction they can selectively choose a different parent utilizing a simple modification to the ETX , specifically modifying the ETX with the calculated θ between two node's directional vectors.

This can be accomplished by each node maintaining the knowledge of their current direction and broadcasting it with their beaconing or by use of some compass or GPS. To limit the amount of data required all nodes would assume the same domain, such as 0° being due east, 90° being north, 180° West, 270° South. Utilizing a simple calculation a node can calculate its own theta based on its current location and its intended location. Formula used is:

$$\theta = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) * 180/\pi \tag{8}$$

This equation provides a θ between 0° and 90° and the vectors are then used to extrapolate the correct θ within a Cartesian plane. When the other node receives this value it will compute its own and will calculate the difference. If the difference is greater than 180 the difference is subtracted from 360° to account for directions that are headed in a similar direction such as 1° North of East and 359° which would actually be 1° South of East.

The idea is that if nodes are heading in the same direction the nodes will stay in range for longer periods of time and thus will require to reselect a new parent less often.

Once the difference of θ is calculated it can be simply summed with the current ETX with the θ, which provides enough adjustment to select a more appropriate node.

$$ETX_{DPCTP\ node} = ETX_{Node} + \theta_{difference} \tag{9}$$

3.3.1.1 Network Initialization

If the network is initialized when all nodes are stationary the first few ETX calculations will not incorporate the Directional Preference modifier as the nodes would not have any displacement and thus could not calculate a θ . The modifier will only be broadcast in the next beacon if the node has moved sufficiently to be able to calculate a valid Δy or Δx. If the nodes are given time to start moving before broadcasting their ETXs the first beacons will contain the modifier $\theta_{difference}$.

3.3.1.2 CTP Data Frame

The data frame for DP-CTP was not altered from the original CTP the definition of it is documented in [14] , the data frame contains the following:

Element	Number of bits	Purpose
P	1	Is the routing pull request. This allows nodes to request routing information.
C	1	Is the Congestion notification. Which is set if a node drops a Data frame.
Reserved	6	
THL	8	Time Has Lived, this indicated the amount of hops this frame has traveled, it is incremented form 0, if the value has reached 255 it will be incremented to 0.
ETX	16	For DP-CTP the ETX is the value calculated in (9
Origin	16	The originating address of the packet, which is never modified once set.
Seqno	8	The originating sequence number, which is never modified once set.

Collect_id	8	Higher-level protocol identifier.
Data	Remainder of packet size	The data being sent, comprised of zero or more bytes.

Table 1: CTP Data Frame

As it is explained in [14] the usage of Origin, Sequno, and Collect_id is used to describe an original data packet that is sent, while using the THL in addition to the previously mentioned items, would describe a unique instance of the data. The ability to determine this information allows for the protocol to recognize a loop and drop packets.

3.3.1.3 DP-CTP Routing Frame

The DP-CTP routing frame format contains the following:

Element	Number of Bits	Purpose
P	1	Is the routing pull request. This allows nodes to request routing information.
C	1	Is the Congestion notification. Which is set if a node drops a Data frame.
Reserved	6	
Parent	8	The node's current parent
ETX	16	For DP-CTP the ETX is the value calculated in (9
Theta	16	the value calculated in (8 which is required to calculate the $\theta_{difference}$ to incorporate in the new ETX

Table 2: DP-CTP Routing Frame

3.3.2 Relative Motion Trickle Algorithm

The Trickle Algorithm originally is used to lower the amount of control overhead that is required for the network to stay healthy. It accomplishes this by increasing the time needed between broadcasting of beacons. The initial time interval starts at some minimum value and doubles after every successful transmission which will increases up to a maximum value [5] [32] . Under certain circumstances the interval must be reset to the minimum value to allow for the connection to stabilize again. The circumstances that induce a reset include the detection of a routing loop, a congested node, or a node broadcasting a beacon with a pull instruction.

This algorithm however was designed with static nodes in mind; as such in a mobile scenario it would not be beneficial at all since all the nodes would have to keep resetting the interval to the minimum value. Ideally updating neighbours with beacons more frequently means that nodes can recover from lost parents more quickly and that disabling the trickle algorithm is better when nodes are moving quickly which was observed by Otman [8].

Since the algorithm is designed for static nodes, and DP-CTP chooses parents based on criteria that would simulate a nearly static network, changing when the trickle algorithm is employed can improve performance. As such DP-CTP enables the trickle algorithm only when a parent that is relatively stationary compared to its child which is determined by calculating a $\theta_{difference}$. If this value is less than or equal to $10°$ the trickle algorithm is enabled. This is beneficial to the network as nodes that are stable update less often while nodes that lose connections frequently are capable of updating faster due to the increased beacons broadcast. Therefore if a parent and its child are traveling in the same direction lowering the frequency of broadcast beacons would decrease overhead, collisions, and power consumption. The reduction in collisions would also lower the frequency of retransmissions possibly lowering the amount of power required to handle communication. The reason $10°$ was chosen was that nodes would stay approximately twice as long within range compared to traveling apart at $20°$.

$$Time\ for\ nodes\ to\ be\ out\ of\ reach = \frac{R}{V\ Sin(\theta)} \tag{10}$$

Utilizing equation 10 using 30 meters for R (range) as the max radio range two nodes would need to travel for, V being the velocity of the nodes at 4m/s and θ being the number of degrees the nodes are traveling apart for the first case $10°$ it evaluates to approximately 43 seconds before the nodes are out of range of each other. Conversely under the same circumstances but traveling $20°$ apart it would take 21 seconds for the two nodes to be out of range. If the trickle algorithm were able to allow for two nodes to increase their update duration sufficiently it would take too long for the nodes to realize they have left each other's radio range.

Chapter 4

Performance of Directional Preference CTP

In this chapter we investigate the differences between CTP and Directional Preference CTP using the same parameters. The performance of DP-CTP is also analyzed with varying parameters such as speed, number of nodes radio range. Additionally DP-CTP is compared with a few differing routing protocols that utilize different topologies. All simulations use random way point for the mobility model. This model implemented in the simulators has nodes that reach the boundaries bounce back into the simulation area.

4.1 Network Selection Criteria

The radio model explained in [8] contains the following equation:

$$PL(d) = PL(d_0) + \mu 10 \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{111}$$

Where:

- PL(d) is the path loss at distance d
- d_0 is a reference distance usually defaulted to a value of 1
- $PL(d_0)$ is the path loss at distance equal to one which is equal to approximately 54dbm
- μ is the path loss exponent
- X_σ is a Gaussian zero mean random variable

As the network and simulator are the same as the one in [8] this equation can be used to determine the distance at different transmission powers. The following table indicates the range of the radios based on transmission power.

Transmission Power (in dbm)	Approximate Range in meters
-7 dbm	25.5 m
-5 dbm	31 m
-3 dbm	37.5 m
-1 dbm	45.5m

The number of nodes selected for the various tests were roughly calculated utilizing

(3 and a varying range of nodes were selected for each of the scenarios to observe the effects different number of nodes in a network can have. For the DP-CTP performance with varying network areas sizes the number of nodes was selected by way of scaling of the 50x50 area and 150x 150 areas to attempt to keep the ratio of nodes to area the same. Additionally Equation (2) was utilized to determine that 60 nodes would allow each node to on average retain at least 10 neighbouring nodes in its neighbour table. This was not changed between simulations as retaining more nodes would have resulted in each node requiring additional time to decide on a new neighbour.

4.2 CTP vs Directional Preference CTP

In this section we investigate the differences between CTP and Directional Preference CTP using the same parameters. Results are the average of 20 repetitions of the same scenarios in order to lower the impact of outlying data points.

The first test scenario was designed to observe how the algorithm performed without the use of the trickle algorithm which is used in static CTP to reduce the amount of beacons sent when the system is stable. Ben Otman [8] indicated that the trickle algorithm hinders performance when nodes are mobile because it is unable to keep up with the frequent link breakages. Figure 4 and Figure 5 indicate what the PRR on average was when the minimum and maximum beaconing interval was set to the value on the x-axis (disabled trickle algorithm). Figure 6 and Figure 7 indicate what the PRR on average was when the minimum beaconing interval was 10000ms and the maximum beaconing interval is the value indicated by the x-axis. Equations 2 and 3 were utilized to determine the number of nodes and radio ranges, equation 2 was used to determine the number of nodes to maintain a node degree of approximately 9 nodes.

Field Layout:	
Length	100m
Width	100m

Number of nodes	60 (uniformly distributed)
Sink location	50,100
Application Layer Parameters:	
Data Traffic	0.333 packets/second
CTP values:	
Min interval	10000
Mobility:	All nodes mobile except sink
Mobility Update interval	100 milliseconds
Speed	1m/s
Move time	30 seconds
Pause time	10 seconds
Random direction change after move time	True
Simulation length	300 seconds
Disabled Trickle Algorithm	

Table 3: CTP vs DP-CTP Test Parameters

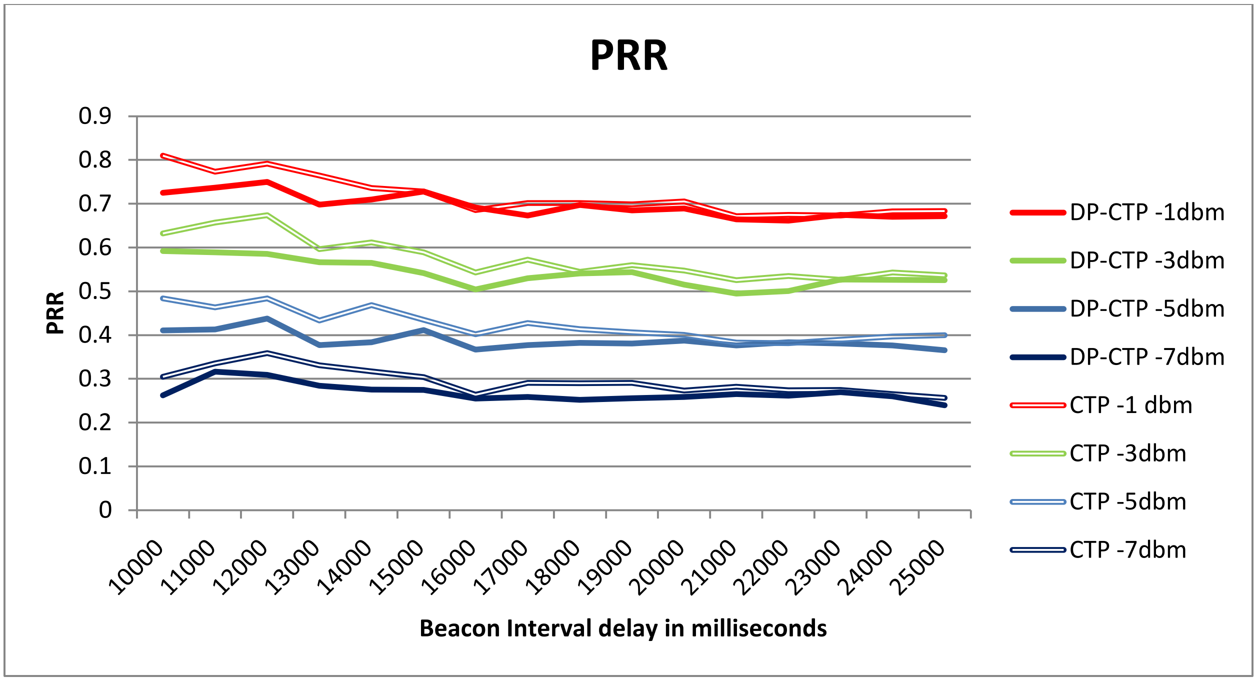

Figure 4: Packet Reception Ratio vs Beacon Interval with Trickle Algorithm Disabled

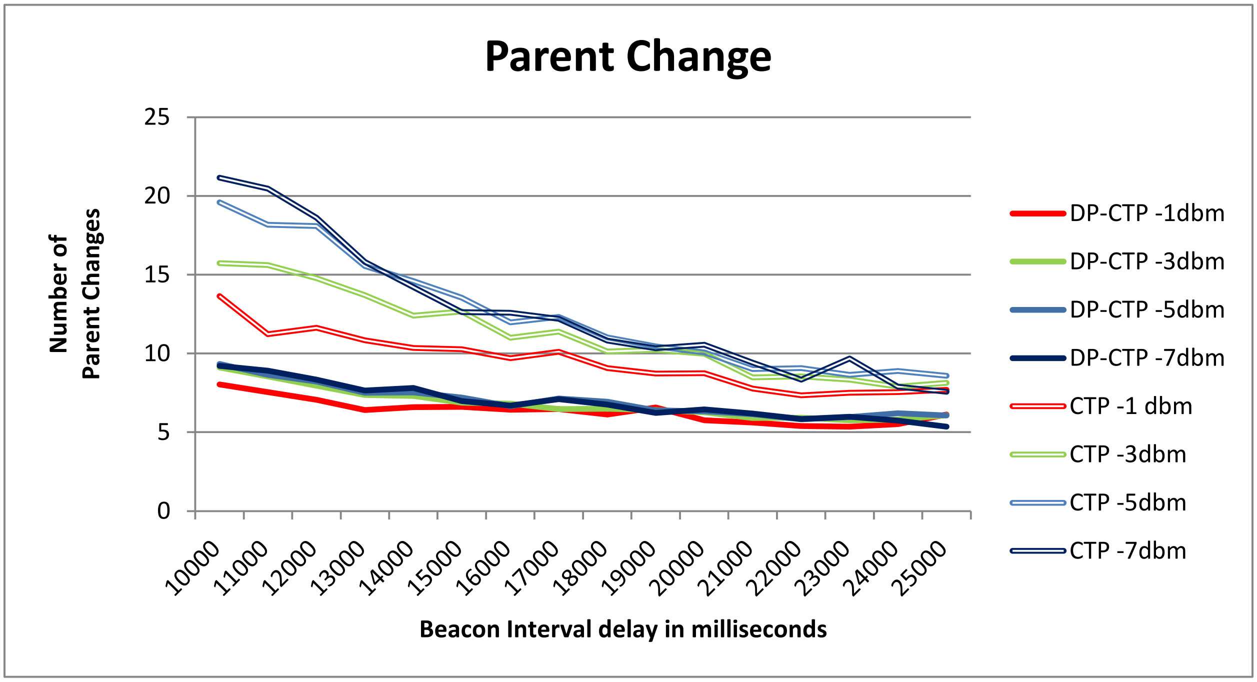

Figure 5: Parent Change vs Beacon Interval Delay

The Packet Reception Ratio (PRR) is based on the total application data sent and received. As it is shown in Figure 4 there is little difference between Mobile CTP and regular CTP regardless of the radio strength. The slight difference that does occur is due to the lack of initial motion at time zero of the simulation. At time zero the nodes do not have a direction as such the default ETX is used to decide on a parent node. This means that at the beginning of the simulation there is a need to change parents more frequently because of a rapidly adjusting ETX, however once the network stabilizes it performs similarly to regular CTP. Figure 5 shows however that the regular CTP has more parent changes on average then Mobile CTP during the entire simulation length of 300 seconds, although this is the case the changes occur throughout the duration of the simulation while Mobile CTP has these changes occur generally at the beginning of the simulation. Mobile CTP has fewer parent changes because the algorithm chooses a parent node that travels in the same or similar direction so the need to change parent is lowered. In addition the amount of parent changes drops as the interval delay increases because it takes longer for the nodes to send out a beacon, which means it will take the nodes longer to select a new parent.

The next simulation utilizes the trickle algorithm, which increases the beacon interval time as packets are successfully received. The minimum beacon interval was selected at 10000 ms because the network needs to update frequently when there is a disturbance, but once it becomes stable decreasing the amount of beacons sent is ideal.

The following figures utilized the same criteria as the previous except that the Trickle Algorithm was enabled

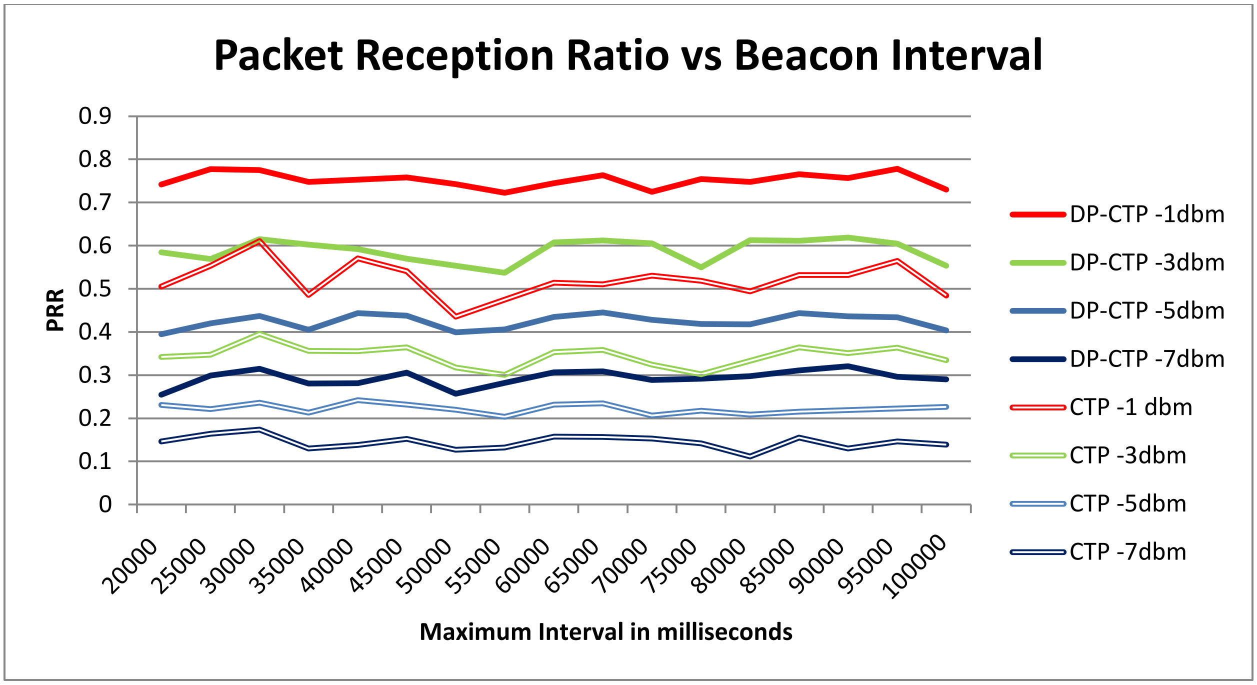

Figure 6: Packet Reception Ratio vs Beacon Interval with Trickle Algorithm enabled

Figure 6 shows the packet reception ratio performance based on different maximum length between beacon broadcasting. In addition it shows the performance with different radio strengths. In general, the higher the radio strength the better the Packet Reception Ratio (PRR) since the radios can connect to nodes further away which reduces the amount of hops required to reach the sink. However the comparison between Mobile CTP (solid lines) with vanilla CTP (hollow lines) it is plain to see that Mobile CTP out performs CTP even with a weaker radio signal such as Mobile CTP-3dbm (approximately 37m) and CTP -1dbm (approximately 45m). Additionally even though it seems in Figure 6 that the change in performance does not change significantly there is a positive trend in both versions of CTP that as the maximum interval in milliseconds increases the better performance until 95000 milliseconds after which point there is a decline in performance.

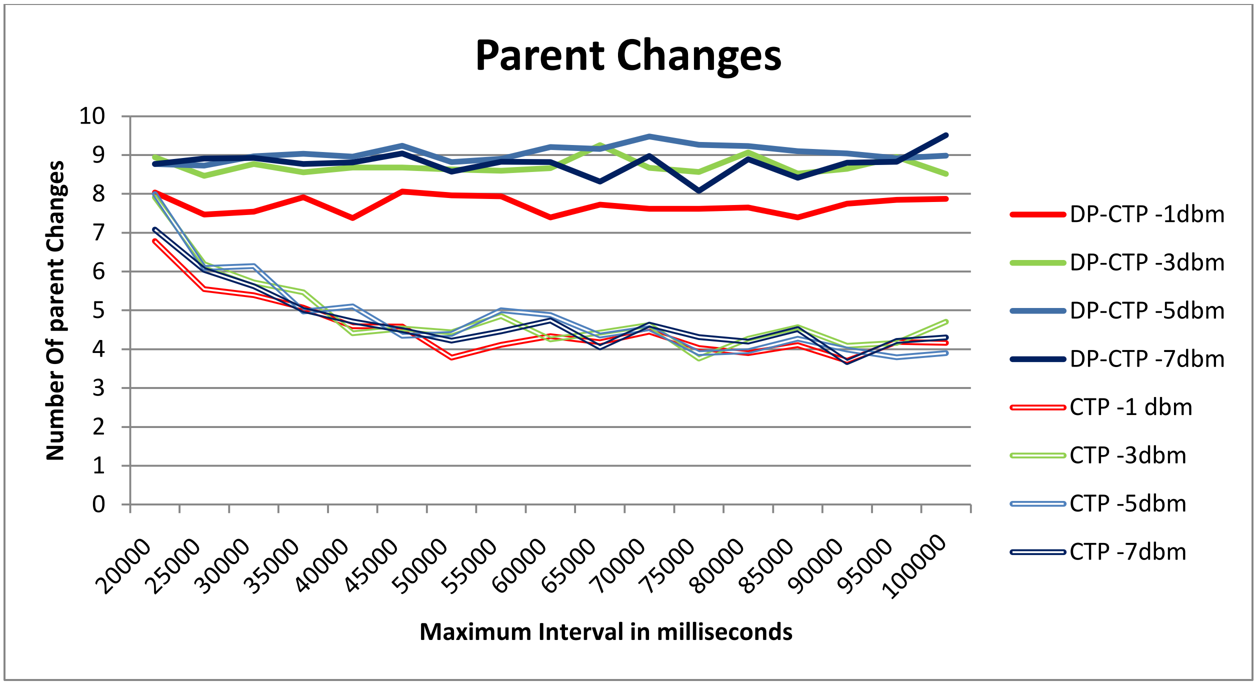

Figure 7: Parent Changes vs Maximum interval in milliseconds

As shown in Figure 7 nodes on average change parents about nine times throughout the simulation length of 300 seconds, consistently regardless of maximum interval length, while regular CTP declines in the amount of node changes the longer the interval becomes. Mobile CTP's lower performance in keeping the same parent node is due to the late start of utilizing directional information because the ETX is modified when the direction is calculated. Subsequent beacons sent cause nodes to change parents because all the newly broadcast ETX values now incorporate this new directional information and make neighbours more attractive parents. This means that additional changes occur at the beginning of the simulation but taper off to lower amounts of parent changes later on in the simulation. The side effect of selecting nodes based on the modified ETX means that there is no longer a situation where one node closer to the sink has the best ETX and is subsequently selected by all its neighbours thus flooding it instantly. The adjusted ETX allows for a more spread out tree to be constructed thus lowering the amount of back off packets sent by congested nodes, as well as lowering the amount of dropped packets in general. It should be noted that the interval of confidence is too small to show on these graphs.

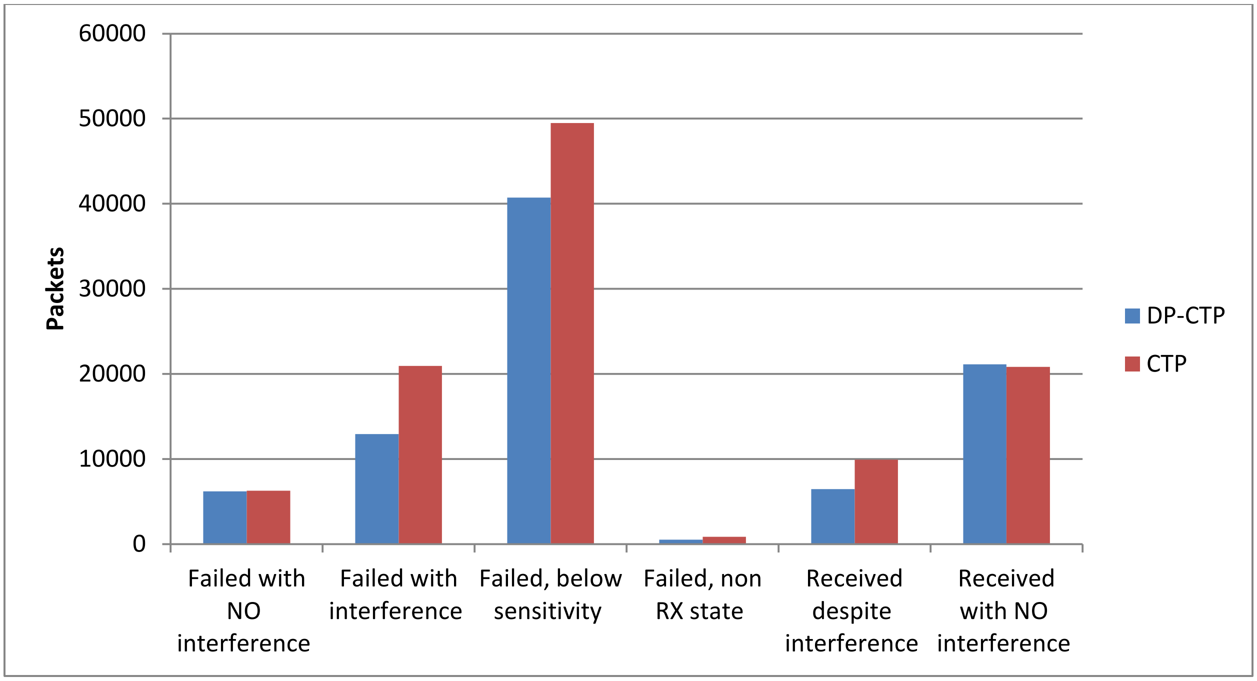

Figure 8: Breakdown of Received Packets

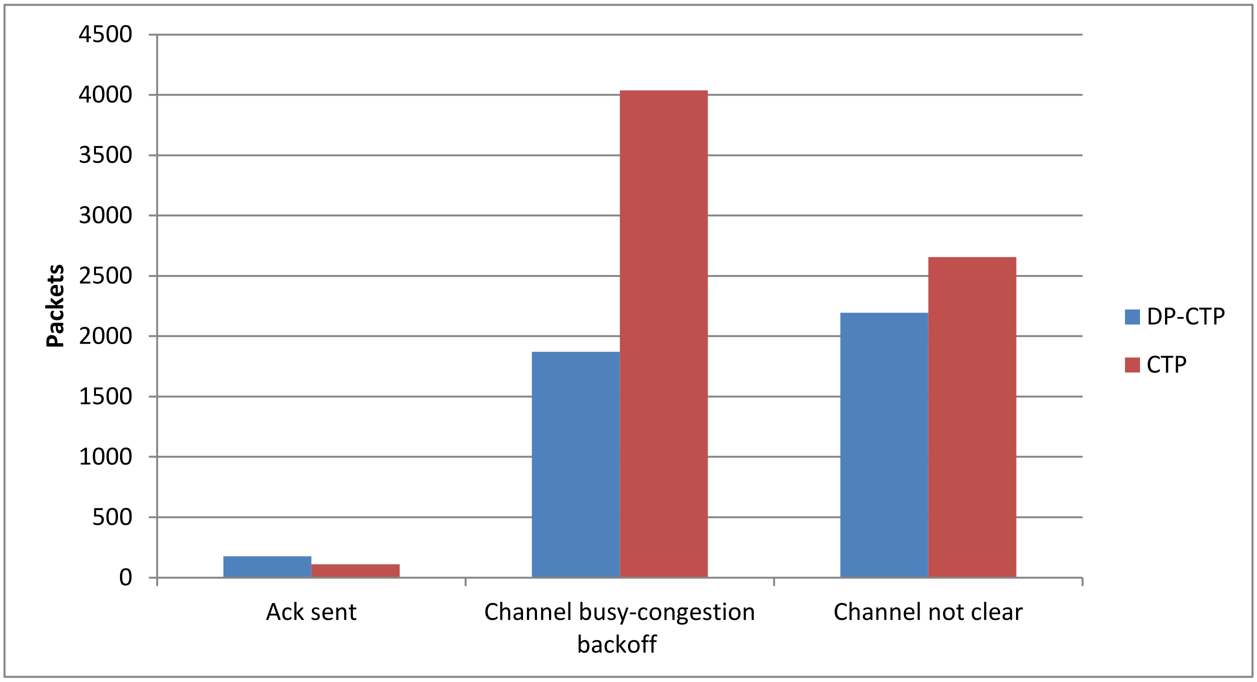

Figure 9: Breakdown of amount of control packets sent

As can be seen by Figure 8 Mobile CTP performs with fewer failed received packets, be it due to low sensitivity or interference. This can be attributed to the dispersion of selected parent nodes around the sink based on the modification of the ETX based on directional preference. This adjustment can also be attributed to a lower count of channel busy-congestion back off control packets broadcast that can be seen in Figure 9. These results can be attributed to the fact that the parent nodes selected are more distributed because there is no longer a single best ETX near the sink, nodes are better or worse as a parent in respect to the child node. This causes nodes to generally select a parent that travels in the same direction over another node which may have a strong ETX otherwise, which in turn lowers the amount of child nodes that are dependent on the same parent node, thereby lowering possibility of congestion or collisions in the vicinity

4.3 Performance Analysis

The following figures and table have been designed to analyze differing performance of DP-CTP protocol based on varying field sizes, node amounts and transmission powers and comparing the results. The following tables describe the specific Simulation Parameters:

Small Field Area

Parameter	**Value**
Field Size	50m x 50m
Amount of Nodes	5,10,15,20,25
Sink Location	x=25 y=50
Frequency of Data Traffic	0.333 packets per second
Transmission Power	-3 dbm
Beacon Intervals:	
Minimum	10000 milliseconds
Maximum	95000 milliseconds

Table 4: Performance Parameters for a Small Field

Medium Field Area

Parameter	**Value**
Field Size	100m x 100m
Amount of Nodes	20,40,60,80,100
Sink Location	x=50 y=100
Frequency of Data Traffic	0.333 packets per second
Transmission Power	-3 dbm
Beacon Intervals:	
Minimum	10000 milliseconds
Maximum	95000 milliseconds

Table 5: Performance Parameters for a Medium Field

Large Field Area

Parameter	**Value**
Field Size	150m x 150m
Amount of Nodes	45,90,135,180,225
Sink Location	x=75 y=150
Frequency of Data Traffic	0.333 packets per second
Transmission Power	-3 dbm
Beacon Intervals:	

Minimum	10000 milliseconds
Maximum	95000 milliseconds

Table 6: Performance Parameters for a Large Field

Only the Field Size, and amount of nodes change between the simulations. The 100 by 100 field area simulation contains the ideal ratio of nodes to area including the radio range utilizing equations 2 and 3, node amounts and field size between the other simulations are incremental ratios to aid in comparison. Sink Location changes only to maintain a location that is at the top center of the field in question. Additionally the ratio x axis of these figures was chosen in order to more easily compare different network areas based on density.

4.4 Packet Reception Ratio Comparison

Figure 10 shows how the Packet Reception Ratio (PRR) that was calculated from the simulated 50m x 50m network area. It is clear that in a small area few nodes can accomplish an extremely good PRR while moving at walking speeds. Although it may seem that there is a wide discrepancy between these speeds a 1.5% difference in ratio change is insignificant. For such a small number of nodes a disconnect between nodes does not affect a very large chain however a single node does comprise a large portion of the network, as such if a node leaves the range of its parent the loss of packets can be the cause of a 1% difference in ratio.

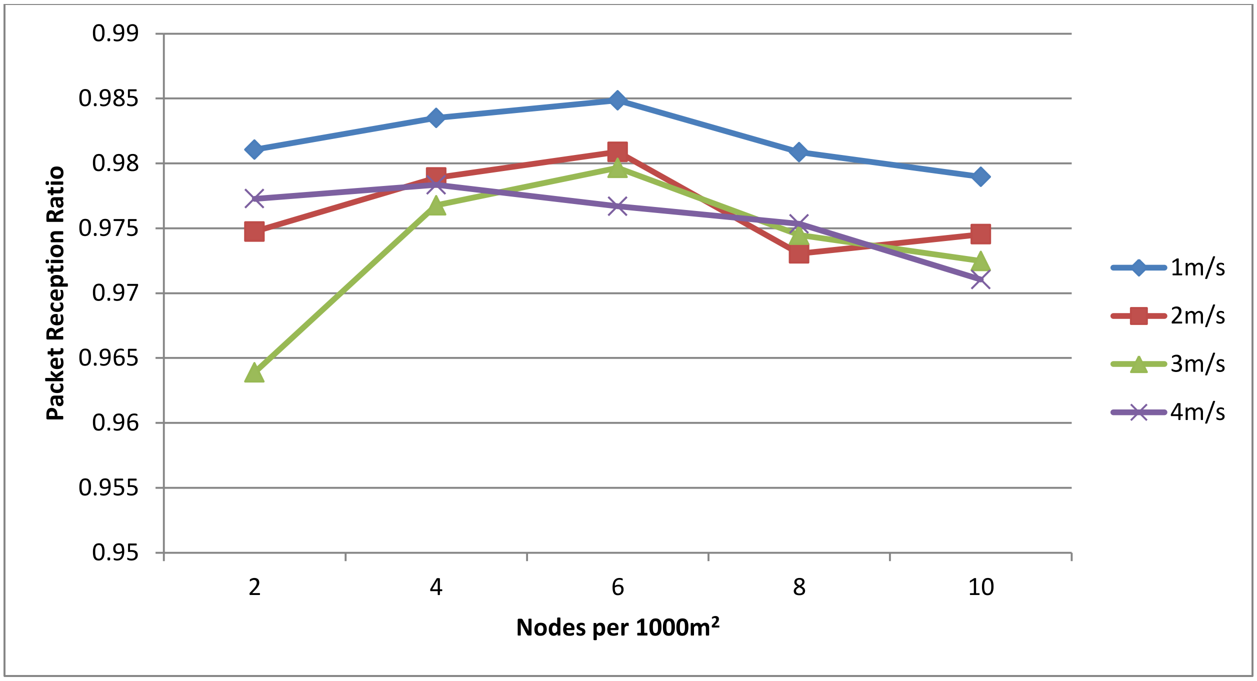

Figure 10: PRR vs Node Coverage in a 50x50 area

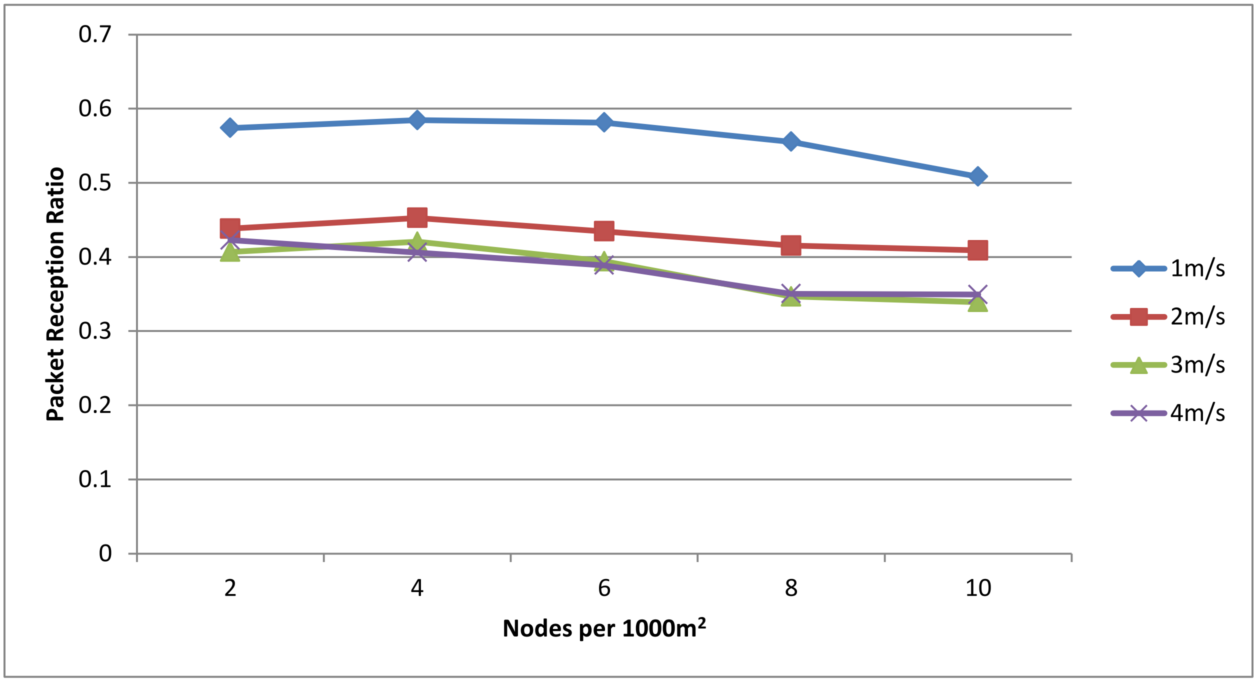

Figure 11: PRR vs Node Coverage in a 100x100 area

Figure 11 shows a consistent PRR in the medium 100m x 100m area with slower moving nodes maintaining the best PRR while faster moving nodes perform marginally worse however perform similarly regardless of the volume of nodes in the network. It is worthwhile to mention that the medium sized network contained 20-100 nodes among the various simulations.

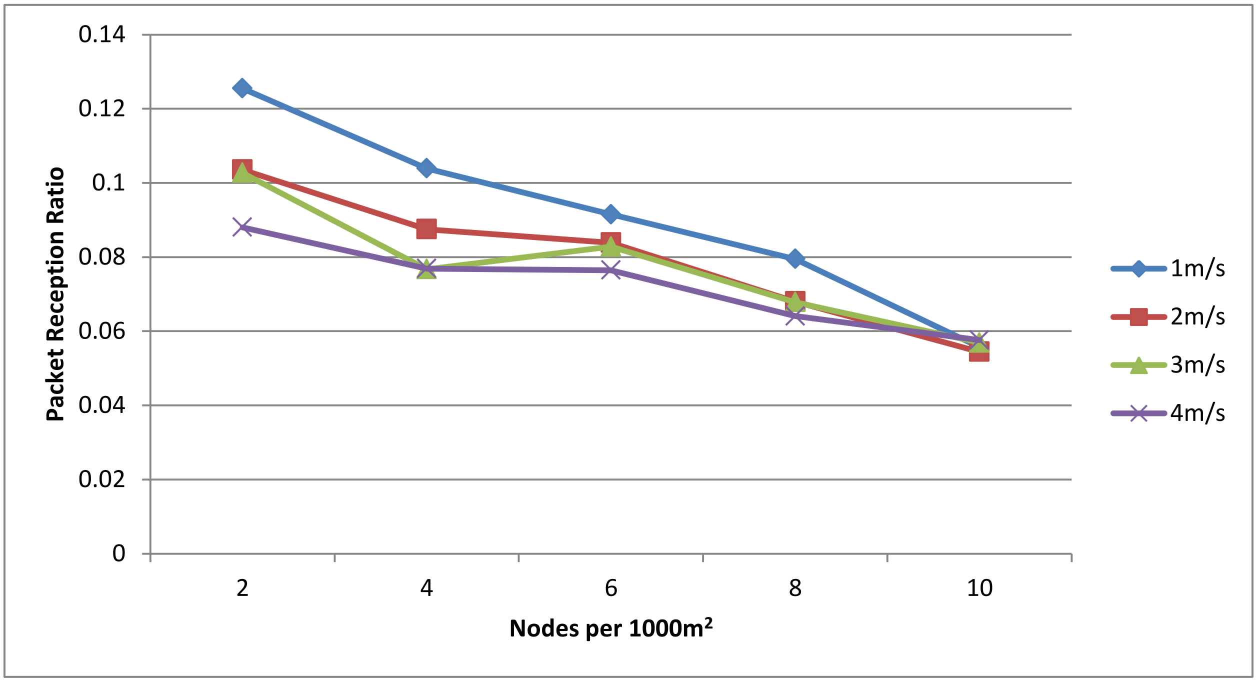

Figure 12: PRR vs Node Coverage in a 150x150 area

While Figure 12 shows the results from the large 150m x 150m area, the algorithm at this scale performs worse in comparison to the small and medium sized networks, but this can be accounted by the number of nodes in the area which were in the range of 45 to 225 nodes.

The low Packet Reception Ratio that can be seen in the large area network can be attributed to the numerous amounts of nodes. Although CTP is capable of accommodating any number of nodes the more nodes that are applied the greater chance there is of collisions and congestion to occur. The increase nodes means that there is an exponential increase in the amount of beacons being broadcast and received as well as packets that are to be forwarded which can account for the increase in collisions as well as the congestion as each node will be listening for beacons and processing each one that it can hear. An additional reason why the large area performed poorly in comparison to the smaller areas is that when a parent closer to the sink in a longer chain of nodes loses its link the rest of the nodes will have lost their ability to forward their packets as well, further lowering the PRR. Following the theme of inability to connect another reason for lower PRR in the larger area can be attributed to nodes that become cut off from the sink and are unable to reconnect because a group of nodes are isolated in a corner of the field. The random movement of the nodes means that there is a change that some nodes may stay isolated for periods of time. The major limiting factor here is the radio range of the nodes, increasing concentration of nodes does not indicate better connectivity but rather additional disruptions. Since the nodes have small radio ranges but must perform in a large area the long chains of nodes that are required to maintain connectivity means that packets send from a child furthest from the sink has that many more chances to be interfered with, transmitted with errors or dropped entirely.

In the small network the radio range is comparatively large compared to the field area which means that the packets have a higher chance to reach the sink, which is evident by the high PRR seen in Figure 10. The medium network performs admirably for the size and quantity of nodes, and is reliant more equally on the routing of the protocol and the radio ranges. Regardless of the volume of nodes both the small and medium networks performed with little to no performance loss as the number of nodes increase in the simulation.

4.5 Parent Selection and Beacon Broadcasting

A quick look at the Beacon broadcast results for the three field size simulations show in Figure 13 the small area network, that the nodes have been broadcasting their beacons excessively especially in comparison to the medium and large area networks Figure 15 and Figure 17 respectively which had much lower beacons broadcast. This can be attributed to an increase in nodes traveling in completely opposite directions after randomly choosing a new direction.

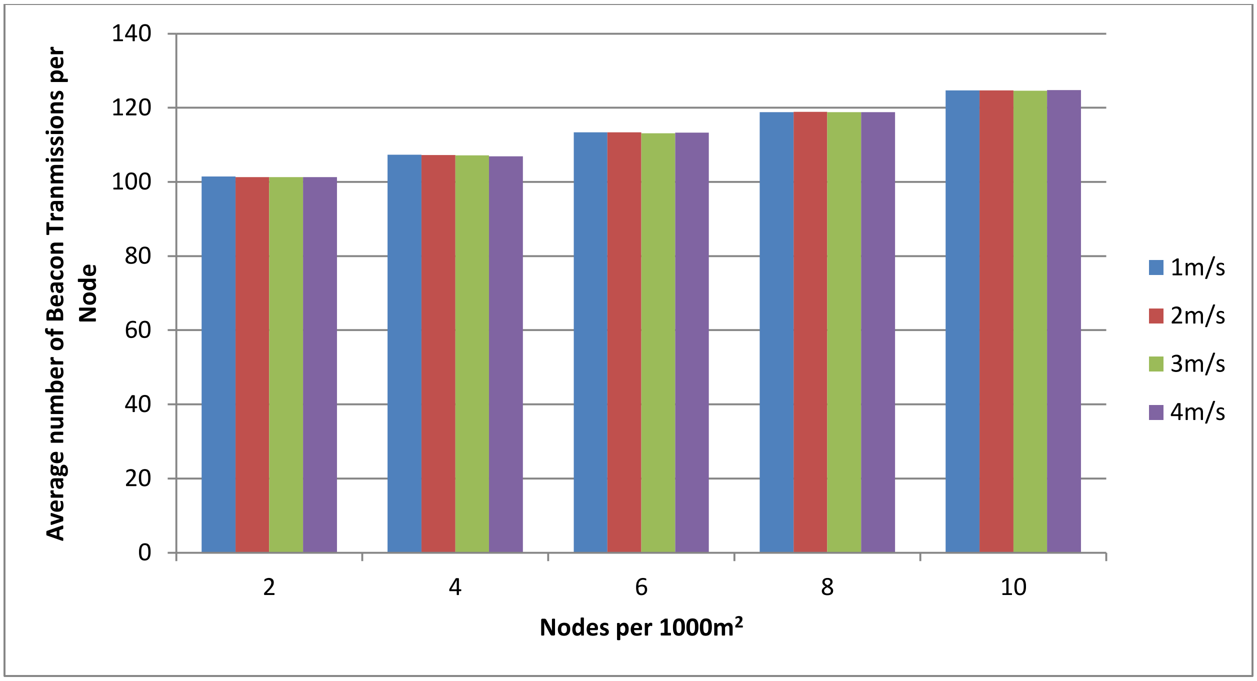

Figure 13: Average number of Beacon Transmissions per node in a 50mx50m area

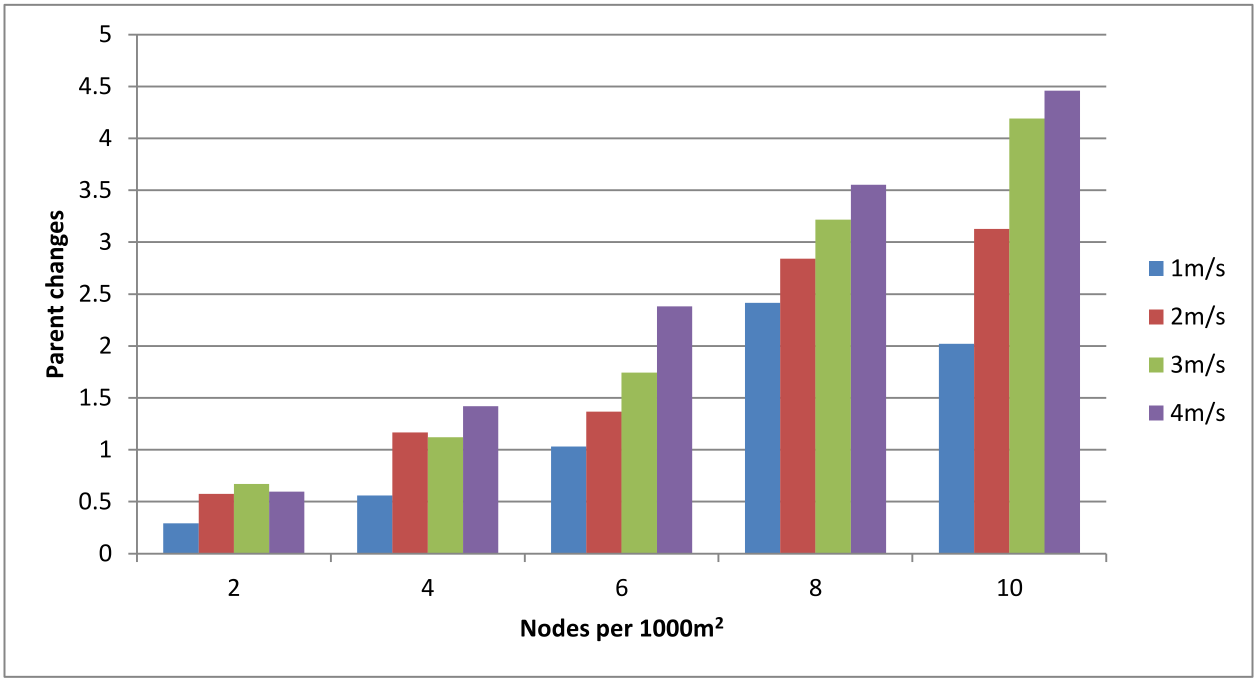

Figure 14: Average Number of Parent Changes per node in a 50mx50m area

The smaller area provides few trajectories to choose from especially since the nodes reach the edges of the field more often, because of this ETX values get updated more frequently which means that parents must update their children with new ETX values. However since there were fewer nodes in the network and the radio range was large enough to generally keep the parent within range there was in comparison very few parent changes, which can be seen in Figure 14. Only at higher speeds were there additional changes in parent however this can be attributed to more frequent directional changes that caused very large changes to the ETX values, sufficient enough to exceed the minimum required improvement to select a new parent. Furthermore since there are fewer neighbours in the small area network there is a high chance that a neighbour with a poor ETX could have replaced the worst case neighbour in the neighbour table. This would mean that the trickle algorithm would have been reset to the minimum interval beacon transmit time more often causing an increase in beacons being broadcast.

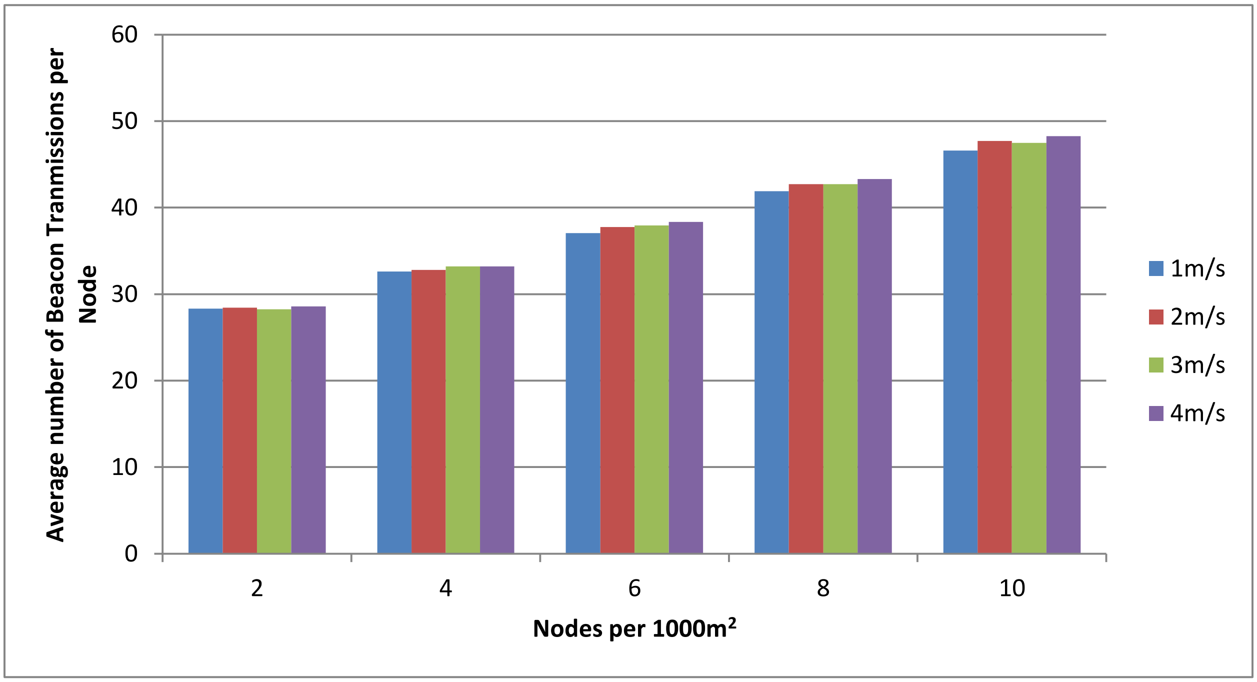

Figure 15: Average number of Beacon Transmissions per node in a 100mx100m area

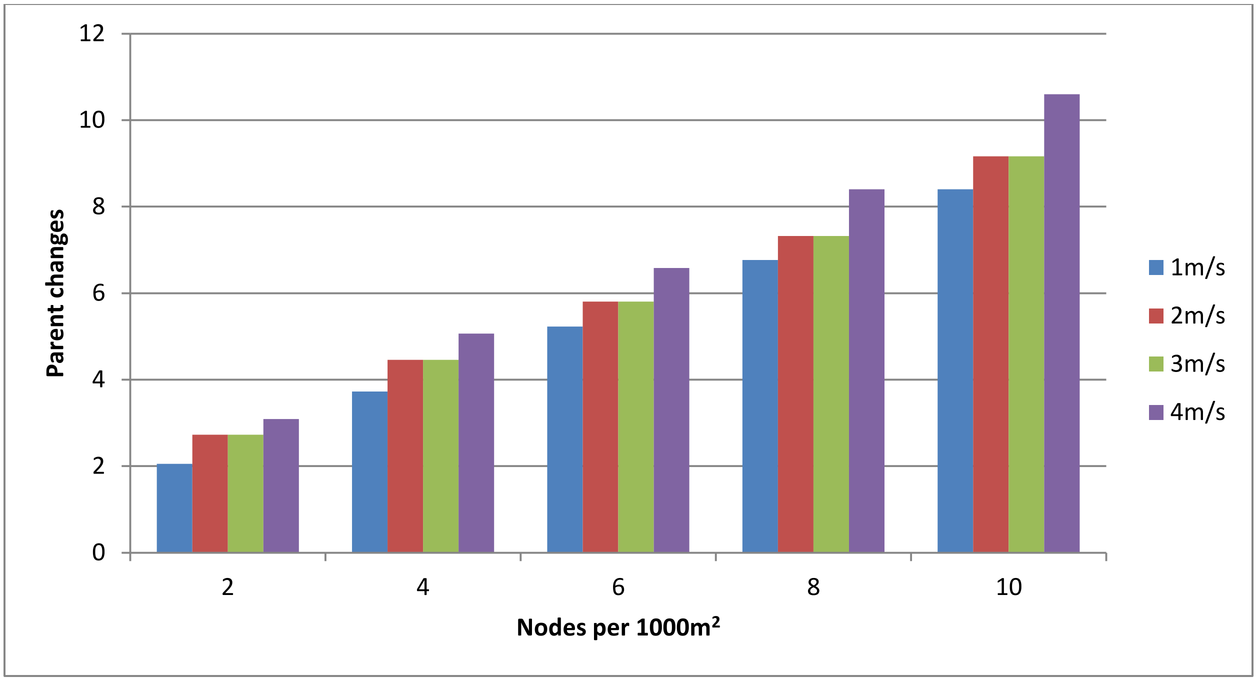

Figure 16: Average Number of Parent Changes per node in a 100mx100m area

Conversely in the medium and large networks, there is many more nodes that can be selected meaning that it would be harder to replace a node in the neighbour table especially if more are selected traveling in the same direction. Since nodes in the larger networks were not being removed from the neighbour table the trickle algorithm had the opportunity to approach their maximum beacons transmit time, thus lowering the amount of beacons broadcast. However due to the larger areas and limited radio range there were more opportunities for nodes to leave and enter the neighbour ranges and possibly replace a nodes parent.

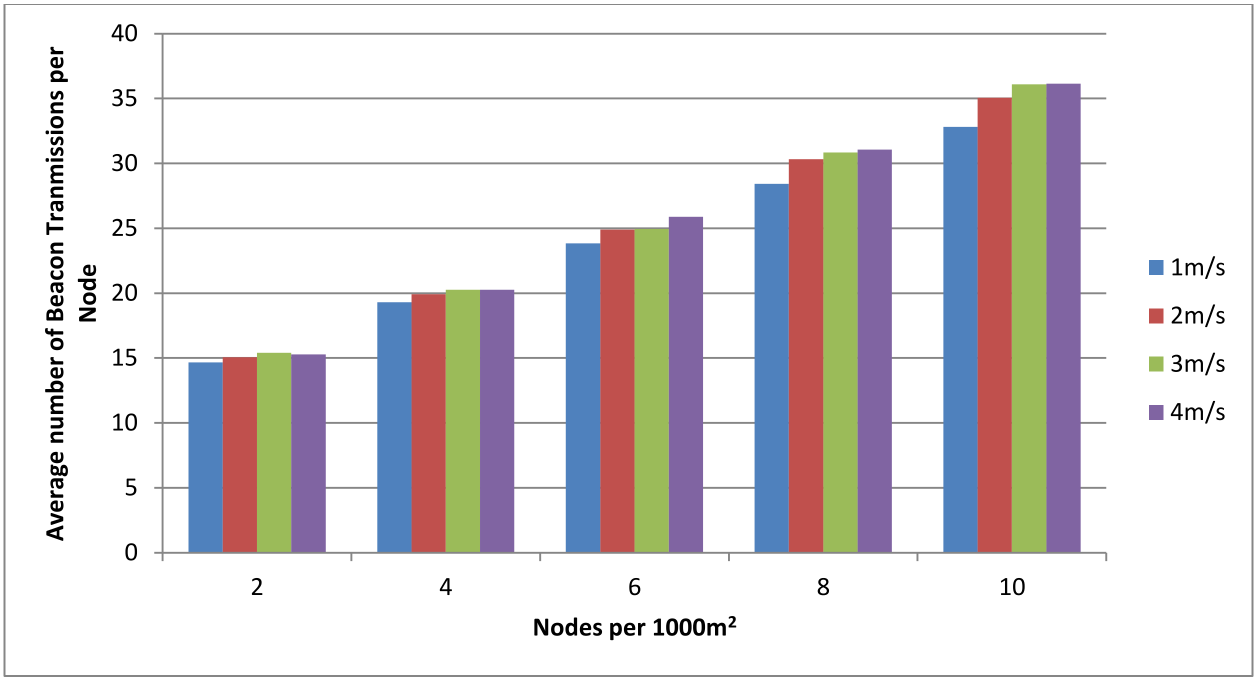

Figure 17: Average number of Beacon Transmissions per node in a 150mx150m area

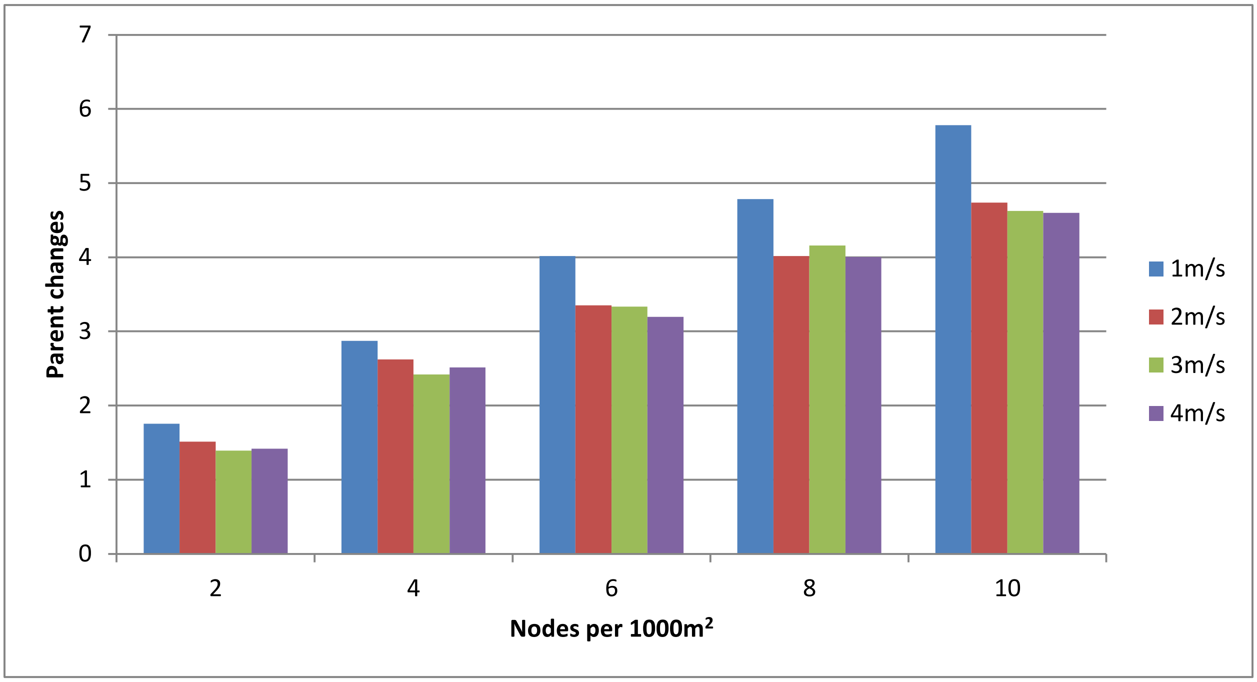

Figure 18: Average Number of Parent Changes per node in a 150mx150m area

4.6 Packet Drops at the Radio Layer

Measuring the performance of packet failures with DP-CTP at the radio layer allows for a cost benefit analysis of the quantity of nodes versus the interference caused by additional nodes and the effect they have on the successful reception of packets. Thermal Noise is calculated by Castalia as the number of packets dropped without interference or bit errors. Figure 19 shows how the increase in nodes lowers the amount of packets failed due to Thermal Noise or bit errors. It is easy to point out that the more nodes in a network there are the lower the thermal noise is a factor in the loss of packets, this is also attributed to the additional interference that occurs with more nodes in a network.

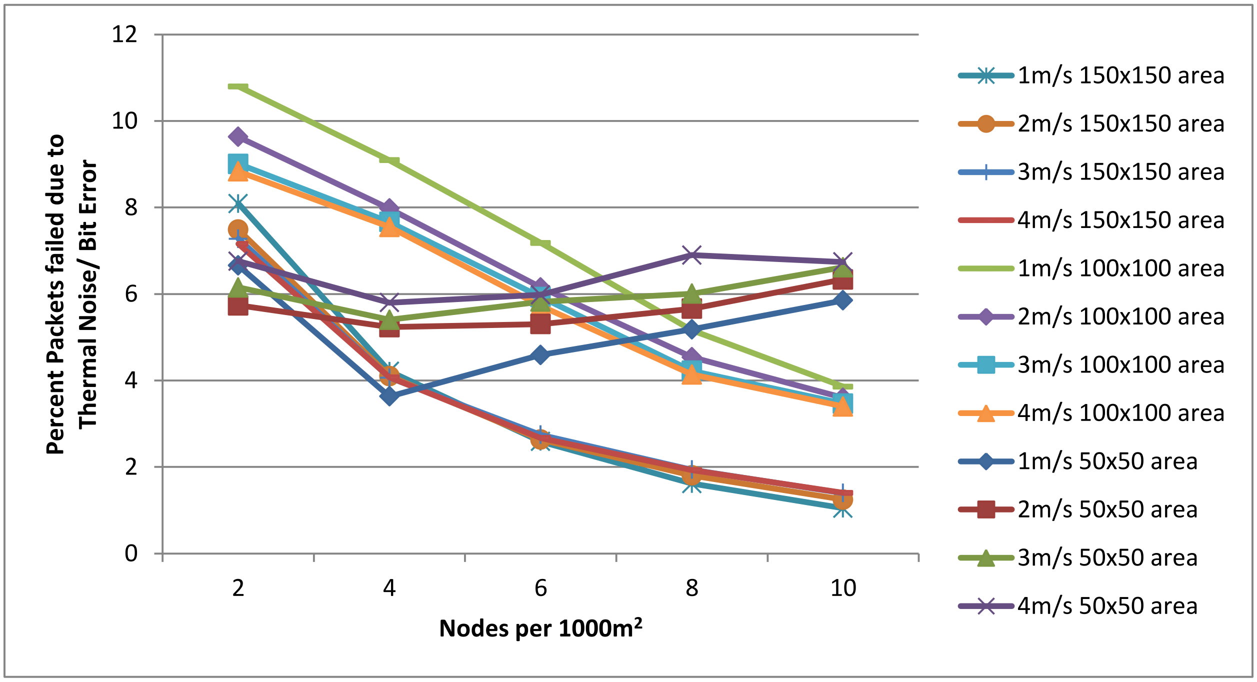

Figure 19: Percent Packets failed due to Thermal Noise/ Bit Error

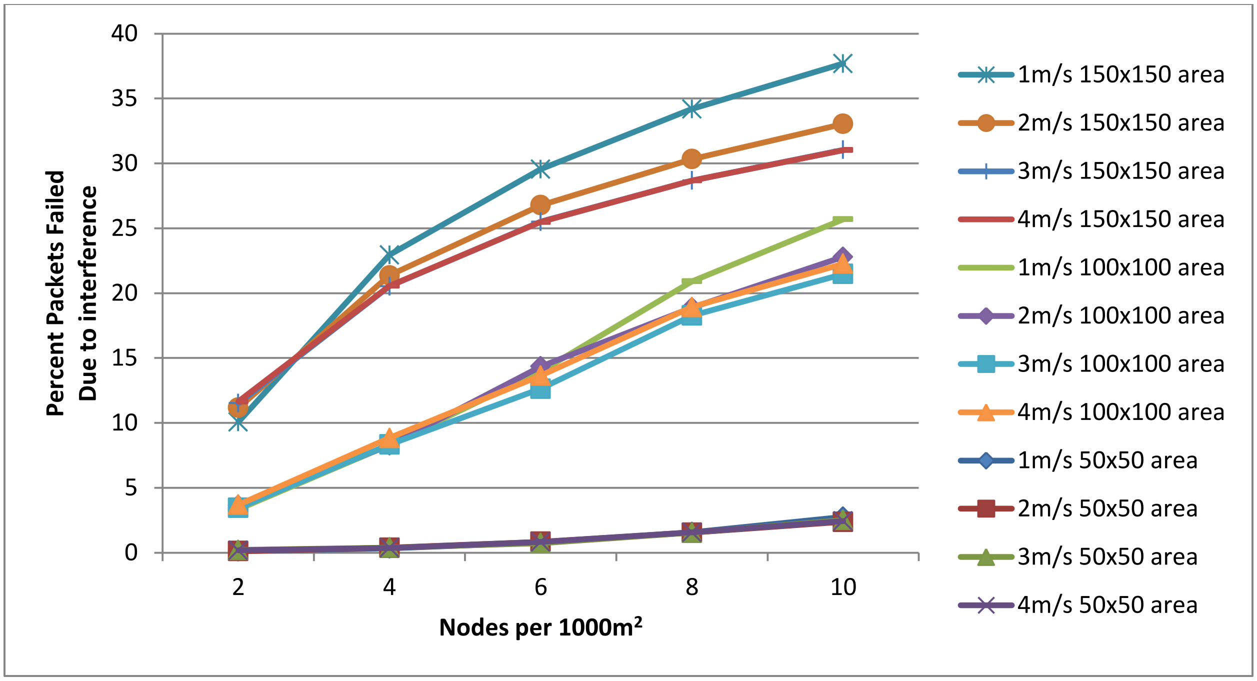

Figure 20: Percent Packets failed due to Interference

As can be seen in Figure 20 however the increase of nodes does mean an increase of interference which regrettably means an increase of failed packets due to interference. Therefore there is a clear trade-off for the quantity of nodes and the amount of packet failures due to interference. However looking at the values of packet failures due to interference for the large area 150x150m area the amount of nodes increased does not have a linear relationship, in fact if the area can support additional nodes i.e. the nodes are more dispersed interference can be mitigated, however this would be at the cost of packet reception ratio as the more nodes there are the longer the paths the packets have to follow the more chances for packets to be lost along the path.

4.7 Packet Drops

Interference and thermal noise are not the sole cause of packet drops, the routing layer contains a few issues that can cause many packets to be dropped, especially when the number of nodes in a network are very high.

Figure 21 shows that the average number of dropped packets due to a busy node increases as the number of nodes increase, in this case the speed is less of a factor as is the number of nodes. The 150x150 area contains 225 nodes at its highest ratio and although nodes further away from the sink would be less busy because they have to forward fewer packets the further they are from the sink. As such the most number of packets dropped occur closer to the sink as the number of possible parents shrink to just a few that are within 1 hop of the sink. It is important to point out that because the value shown below is the average it does not mean that the all the nodes experienced the same number of packets lost due to being busy, but that the number of dropped packets increases the closer to the sink a node is.

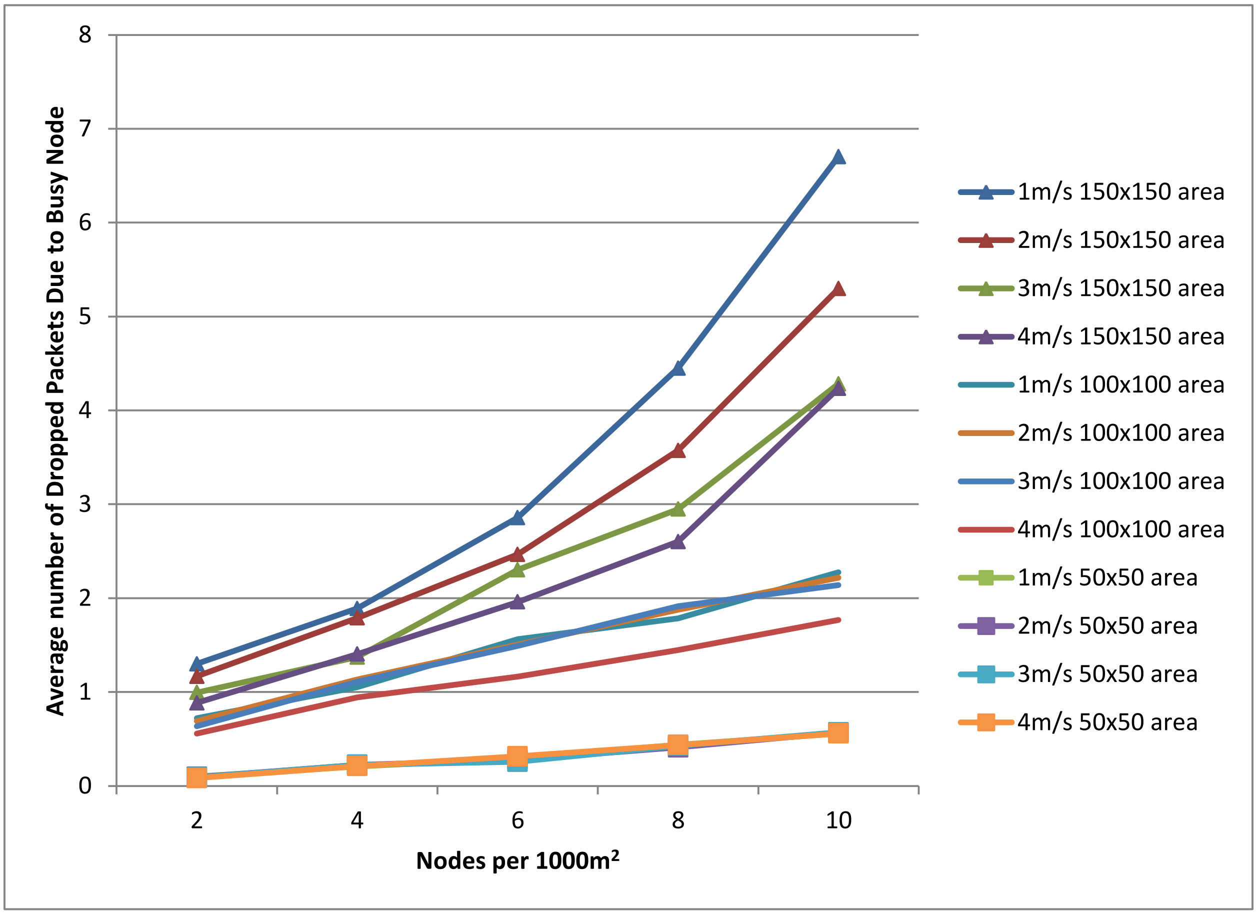

Figure 21: Average Number of Dropped Packets Due to Busy Node

It is evident that on the largest area with the most number of nodes speed does have an impact however the slower speed has the most number of dropped packets, this could be attributed to a more stable tree meaning more packets successfully getting close to the sink meaning additional congestion. Note for these simulation on average each node sends 100 data packets, and the average number of dropped packets number less than 7.

Figure 22 shows that the number of dropped packets increases as the density increases. This is mainly caused by increased interference as more nodes exist the more the packets are broadcast which in turn means more collisions occur. This is more evident with the larger 150x150 meter area where approximately 40 packets on average are dropped. Meaning the denser networks suffer from increased interference, as well as overflowed buffers due to the large throughput of the network. In the case of the 50x50m area the number of nodes is low enough such that interference plays a minimal role, and the volume of the buffers is larger than the throughput.

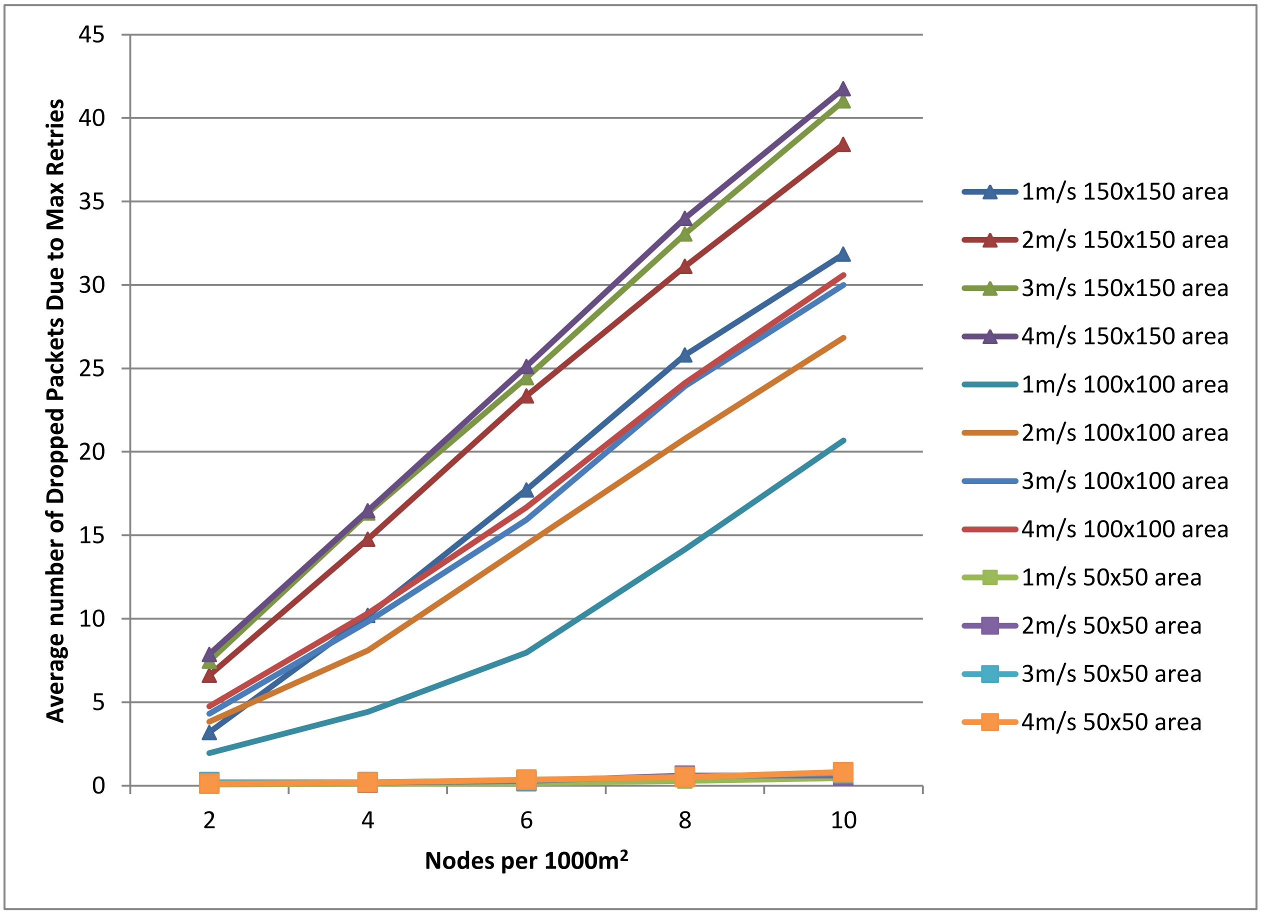

Figure 22: Average Number of Packets Dropped Due to Maximum Retries

The number of times congestion back offs occurred throughout the simulations can be seen in Figure 23 which highlights the issue of a large throughput of data affecting the network. There is an obvious strong correlation of the number of nodes and the affect it has on the buffers of the nodes, specifically nodes that are 1 hop from the sink which are responsible for all of the data packets that are supposed to reach the sink. In the case of the larger networks 1 hop is not the only choke point, as the volume of packets sent can even cause congestion bake-offs to occur just as often at 2 and 3 hops from the sink. Since the nodes change their parents less often data has transmitted has a steady path to the sink which causes the buffers to fill and causes the back offs. Additionally the nodes are distributed uniformly across the field, which means at the beginning of the simulation there is no concentration of communication, however with mobility the nodes tend to gather in small areas for a portion of time which causes mass beacon and data packets to be broadcast which affect any node in the vicinity. Even if a node is not the intended target of the packet, it will still be collected and analyze to determine what is to be done with it. Congregations of nodes in this fashion can cause mass congestion back-off commands to be broadcast.

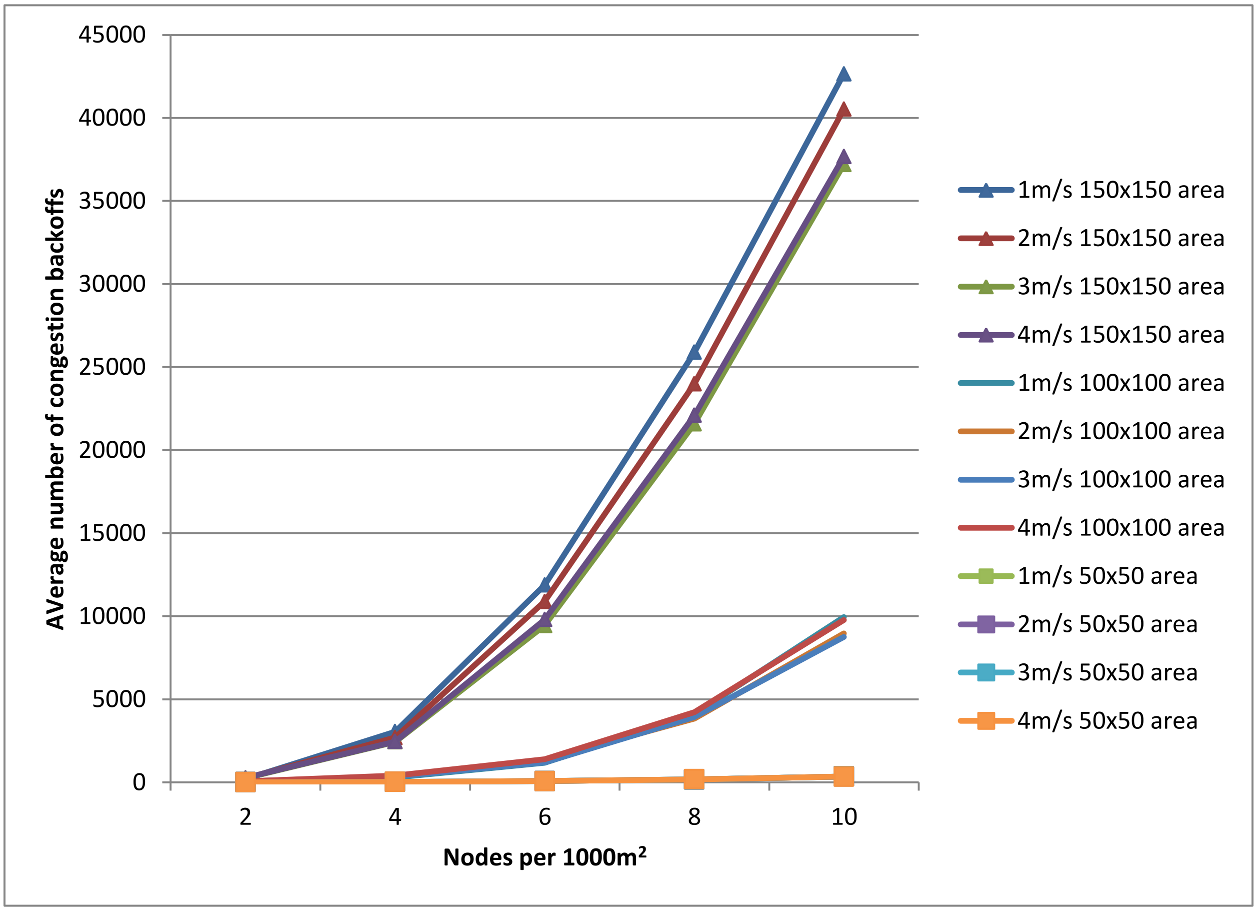

Figure 23: Average Number of Congestion Backoffs

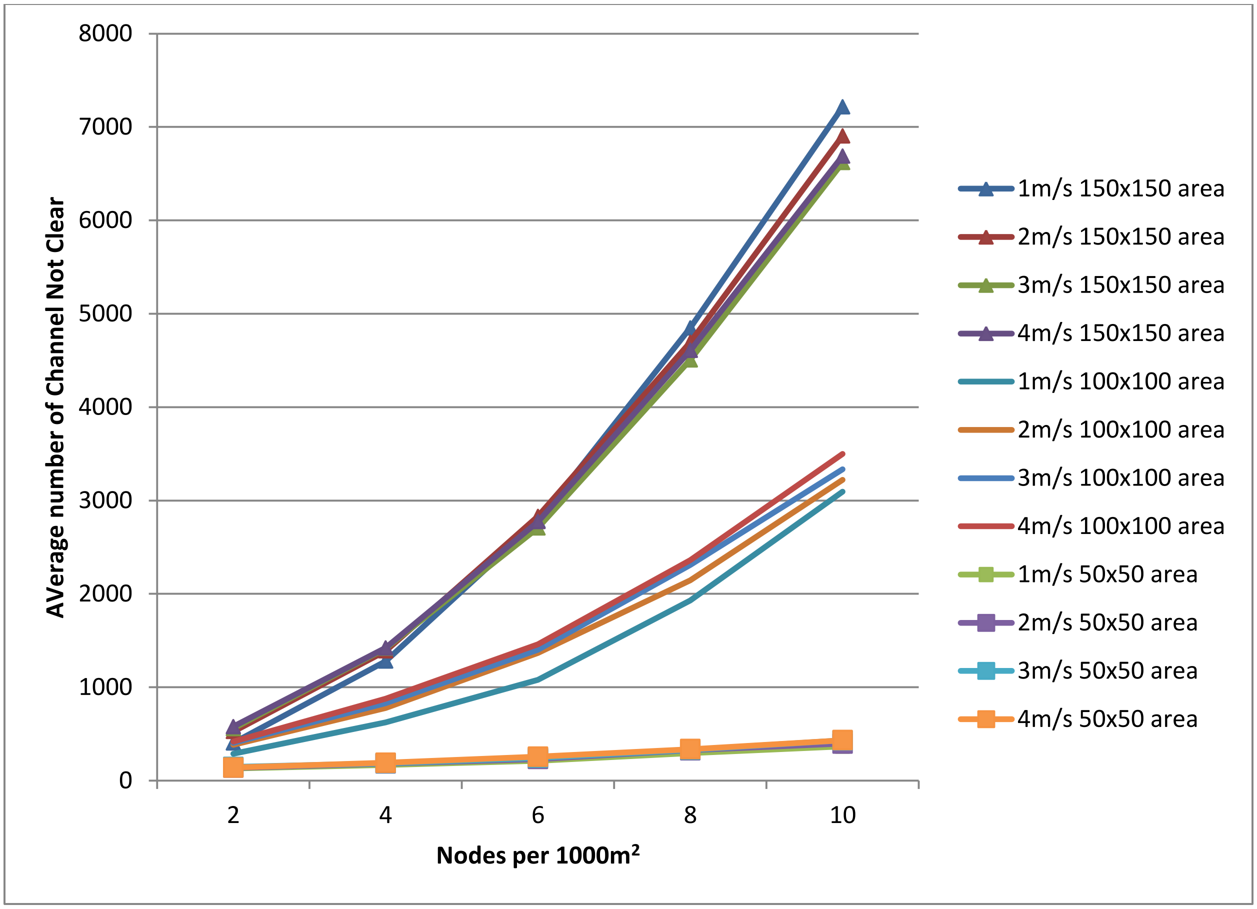

Figure 24: Average Number of Channel Not Clear Responses

4.8 Comparison with M-Leach and GGF

Looking at a simple comparison between some of the more popular protocols utilizing different topologies M-Leach, DP-CTP and GGF in Figure 25, Figure 26, and Figure 27 it can be seen that DP-CTP outperforms the other algorithms in terms of Packet Reception Ratio. Looking at M-LEACH the PRR that it experiences is quite low, this is not usually the case for M-LEACH however M-LEACH is not a multihop protocol as such it is limited to the range of the cluster head and sink, assuming that the cluster head is 30m away from the sink the furthest this network can reach is 60m from a sensing node, this means that there are numerous nodes in an area that are unable to send their payloads. This affects the PRR because it is based on the expected number of packets to be sent versus the total number of unique packets received by the sink. The variation in the slightly higher speeds and PRR that is received for M-LEACH can be attributed to nodes moving into range of a cluster head to deliver their payload.

Continuing the comparison, GGF was run in two different styles, the first where all nodes were mobile except the sink and the second where all the nodes were static while the sink was mobile. The first method where all nodes were mobile while the sink was static had poor PRR, this could be attributed to the fact that each node had to constantly update the location of its neighbours to determine to whom to forward, and because the sink was broadcasting the same location the further away a node was from the sink the less exact it could determine its own location. When the sink was mobile but all nodes were static the PRR performance was much improved, again this can be attributed to the fact that the sink could more rapidly update the nodes in the network with its location and because the nodes are static they were able to better pin point their own locations and not have to update their neighbour table compared to a volatile scenario.

The purpose of comparing DP-CTP against these other popular protocols that utilize different topologies was to observe how well DP-CTP fared against other established protocols that tackled the problem of mobility in WSNs.

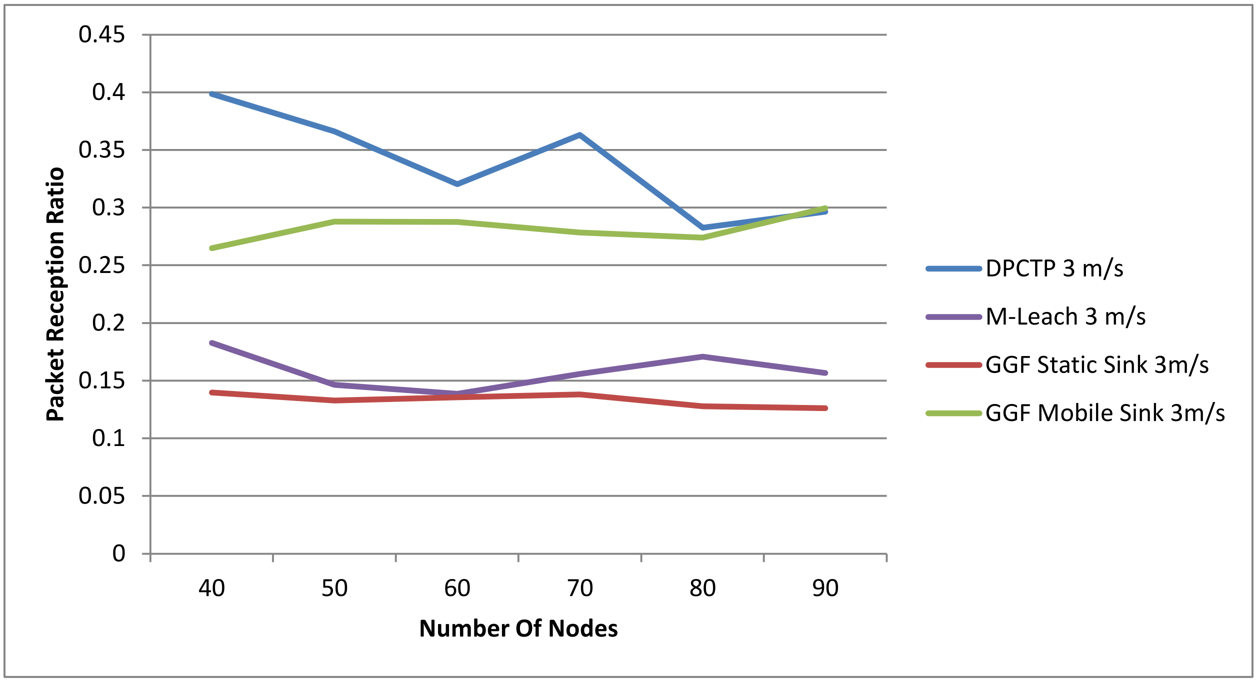

Figure 25: Packet Reception Ratio Comparison 3m/s

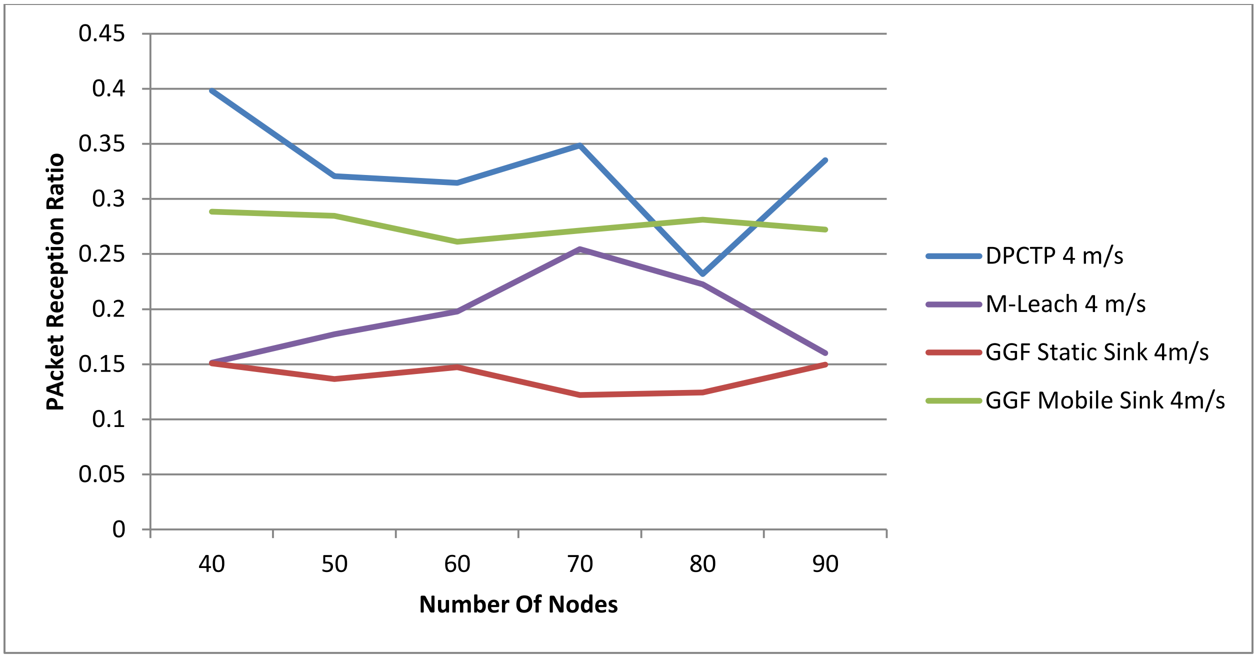

Figure 26: Packet Reception Ratio Comparison 4m/s

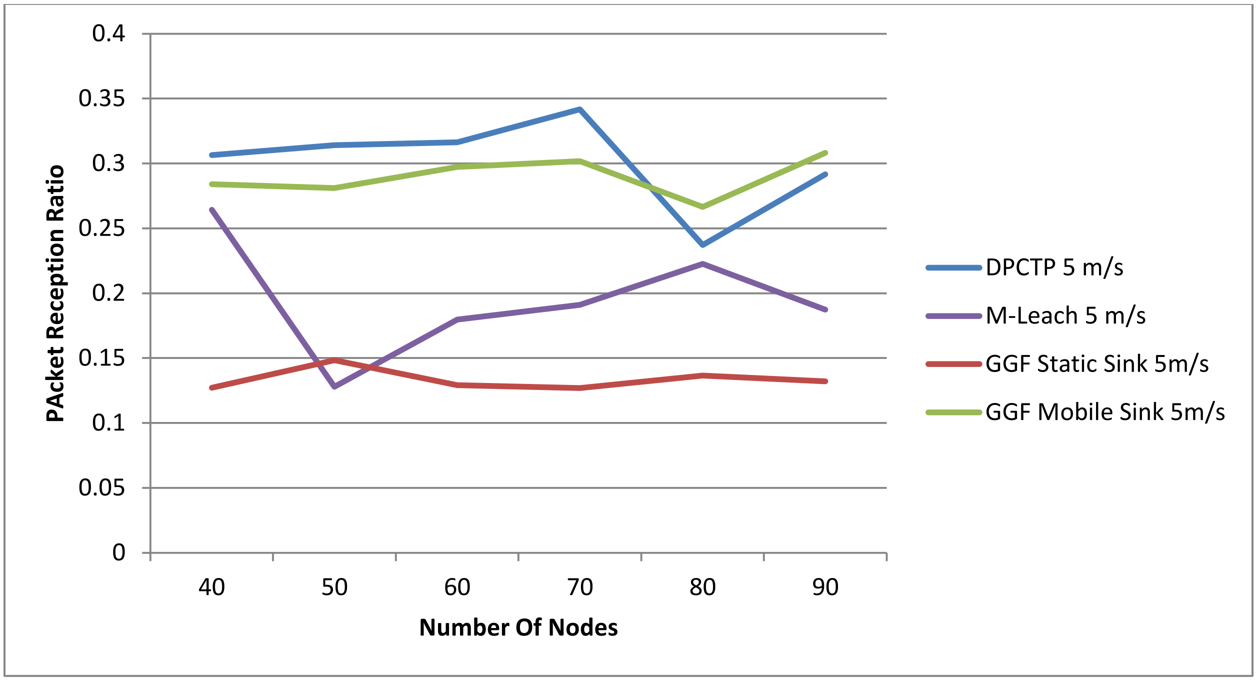

Figure 27: Packet Reception Ratio Comparison 5m/s

Chapter 5

Conclusion and Future Work

In this thesis Directional Preference Collection Tree Protocol was created which is a modification to the CTP protocol. It was created in an attempt to improve the performance of a multihop algorithm by way of improving the longevity of connections in a network and reduce the number of times branches in the tree required rebuilding.

A simple performance analysis was conducted on CTP and DP-CTP to compare the performance of the two algorithms. The results from the simulations showed that DP-CTP improved the performance of the network by lowering the number of parent changes which allowed for the network to stay stable for longer, the tree structure however required an initial stabilization period while the nodes learned what their directions were. This also improved the packet reception ratio of the network as nodes could focus on forwarding data for longer instead of searching for new parents. It was also found that the modifications to DP-CTP allowed for the trickle algorithm to keep beacon updates to a minimum by setting the beacon intervals higher values which reduced overhead and allowed for less interference.

Additionally performance analysis of DP-CTP was conducted. It was determined that although DP-CTP performs better than CTP, there are limitations such that as the volume of nodes in the network increases performance decreases, there is for each network area a number of nodes that is ideal, too few nodes and there is not enough coverage, to many and the interference caused by the additional noise can prevent successful communication. It was also concluded that there can occur an issue with large volume of nodes and the number of dropped packets due to busy nodes, the nodes that are closer to the sink experience large number of packets to be forwarded and the buffer that holds them is incapable of collecting and forwarding them in a quick enough time. As such either the buffer should be increased, or an aggregation system should be implemented.

A comparison against M-Leach was also conducted. M-Leach utilizes scheduling to prevent or lower the effects of collisions and interference, and it was shown that the packets that are sent by the radio more of them are successfully received by their cluster heads. However because M-Leach does not support multihop the application layer reports fewer data packets received by the sink. There is a trade-off between Tree based algorithms and clustering algorithms such as M-Leach and DP-CTP. For clustering algorithms in order to have complete coverage of an area radio range is the limitation which required additional power usage. Tree based algorithms have a limitation to the number of nodes they can directly support due to the number of packets that need to be forwarded especially near the sink where nodes can be overloaded, an increase in buffer can allow for fewer packet drops but would again require additional power. In both cases in order to accommodate larger areas the algorithms require different changes to perform better both of which would result in an increase in power consumption. If however DP-CTP aggregated data then the number of packets sent would be reduced thus possibly reducing the congestion occurring near the sink, while not having a large impact on power consumption.

For future work and development of DP-CTP, comparison and analysis utilizing energy as a metric would be worthwhile, this thesis utilized PRR as various simulators were used to compare various protocols and not all simulators calculated energy in the same manner.

References

[1] D. Sharma, “Evaluating and Improving Collection Tree Protocol in Mobile Wireless Sensor Network,” University of Ontario Institute of Technology, 2011.

[2] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, “A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements,” *IEEE Access*, vol. 5. pp. 1872–1899, 2017.

[3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Comput. Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[5] B. Djamaa and M. Richardson, “Optimizing the Trickle algorithm,” *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 819–822, 2015.

[6] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane, and L. Azouz Saidane, “A survey on fault tolerance in small and large scale wireless sensor networks,” *Comput. Commun.*, vol. 69, pp. 22–37, 2015.

[7] S. Sajadian, A. Ibrahim, E. P. De Freitas, and T. Larsson, “Improving connectivity of nodes in mobile WSN,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 364–371, 2011.

[8] N. Ben Otman, “Design Considerations for Fixed Node Assisted Multi-hop Mobile Sensor Network . By,” University of Ontario Institute of Technology, 2015.

[9] D. Krynicki and R. Liscano, “Directional Preference Collector Tree Protocol for Mobile Wireless Sensing.,” in *Advances in Network-based Information Systems*, 2017, vol. 7, pp. 339–350.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, vol. vol.1, p. 10.

[11] G. S. Kumar, V. P. Mv, and K. P. Jacob, “Mobility metric based LEACH-Mobile protocol,” *Proc. 2008 16th Int. Conf. Adv. Comput. Commun. ADCOM 2008*, pp. 248–253, 2008.

[12] Y. Oh and K. Lee, “A Clustering Algorithm Based on Mobility Properties in Mobile Ad Hoc Networks,” *Int. J. Distrib. Sens. Networks* , vol. 11, no. 6, Jun. 2015.

[13] B. Z. Hakoura, “Comparison of Collection Tree Protocols with Gossip Algorithms for Data Aggregation in Wireless Sensor Networks,” McGill University, 2011.

[14] R. Fonseca, O. Gnawali, K. Jamieson, and S. Kim, “The collection tree protocol (CTP),” 2006.

[15] D. S. J. De Couto, “High-Throughput Routing for Multi-Hop Wireless Networks,” Massachusetts Institute of Technology, 2004.

[16] N. Hassanzadeh, “Scalable Data Collection for Mobile Wireless Sensor Networks,” School of Electrical Engineering Kungliga Tekniska Högskolan, 2012.

[17] J. Flathagen, E. Larsen, P. E. Engelstad, and O. Kure, “O-CTP: Hybrid opportunistic collection tree protocol for Wireless Sensor Networks,” in *Proceedings - Conference on Local Computer Networks, LCN*, 2012, pp. 943–951.

[18] I. Leontiadis and C. Mascolo, “GeOpps: Geographical opportunistic routing for vehicular networks,” *2007 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WOWMOM*, 2007.

[19] J. Zhu, J. Cao, M. Liu, Y. Zheng, H. Gong, and G. Chen, “A mobility prediction-based adaptive data gathering protocol for delay tolerant mobile sensor network,” *GLOBECOM - IEEE Glob. Telecommun. Conf.*, pp. 730–734, 2008.

[20] A. Nasipuri and K. Li, “A directionality based location discovery scheme for wireless sensor networks,” 2002, pp. 105–111.

[21] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, “Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks,” *Proc. 2nd Int. Conf. Embed. networked Sens. Syst. - SenSys ’04*, p. 108, 2004.

[22] M. Bertanha and R. W. Pazzi, “LOGR: Joint LOcalization and Geographic Routing-based Data Dissemination in Wireless Sensor Networks with Mobile Sinks.,” *DIVANet@MSWiM*, pp. 83–90, 2016.

[23] M. Jevtić, N. Zogović, and G. Dimić, “Evaluation of Wireless Sensor Network Simulators,” *Proc. 17th Telecommun. Forum TELFOR 2009 Belgrade Serbia*, pp. 1303–1306, 2009.

[24] H. Sundani, H. Li, and V. Devabhaktuni, “Wireless Sensor Network Simulators A Survey and Comparisons,” *Int. J. Comput. Networks*, vol. 2, no. 2, pp. 249–265, 2010.

[25] A. Stetsko, M. Stehlík, and V. Matyas, “Calibrating and comparing simulators for wireless sensor networks,” *Proc. - 8th IEEE Int. Conf. Mob. Ad-hoc Sens. Syst. MASS 2011*, pp. 733–738, 2011.

[26] A. Boulis, “Castalia A simulator for Wireless Sensor Networks and Body Area Networks - User’s Manual,” *Version 3.0*, 2010. [Online]. Available: http:/ /castalia.npc.nicta.com.au/.

[27] T. D. Lagkas, A. Lamproudi, P. Sarigiannidis, and C. Skianis, “The impact of mobility patterns on the efficiency of data forwarding in MANETs,” *IEEE Int. Conf. Commun.*, vol. 2015–Septe, pp. 6037–6042, 2015.

[28] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 483–502, 2002.

[29] F. Xue and P. R. Kumar, “The Number of Neighbors Needed for Connectivity of Wireless Networks,” *Wirel. Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[30] P. Santi and D. M. Blough, “The critical transmitting range for connectivity in sparse wireless ad hoc networks,” *IEEE Trans. Mob. Comput.*, vol. 2, no. 1, pp. 25–39, 2003.

[31] U. Colesanti and S. Università, “A Performance Evaluation Of The Collection Tree Protocol Based On Its Implementation For The Castalia Wireless Sensor Networks Simulator,” *Science (80-.).*, pp. 1–47, 2010.

[32] P. Levis, T. Clausen, O. Gnawali, and J. Ko, “The Trickle Algorithm,” 2011. [Online]. Available: http://www.rfc-editor.org/info/rfc6206.