TECHNICAL UNIVERSITY OF CRETE
ELECTRONIC AND COMPUTER ENGINEERING DEPARTMENT
TELECOMMUNICATIONS DIVISION

Multiple Frequency Medium Access Control Protocols And Applications

by

Nikolaos Agadakos

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DIPLOMA DEGREE OF

ELECTRONIC AND COMPUTER ENGINEERING

November 2012

THESIS COMMITTEE

Assistant Professor Aggelos Bletsas, *Thesis Supervisor*
Assistant Professor George N. Karystinos
Assistant Professor Antwnios Deligiannakis

Abstract

In Wireless Sensor Networks (WSN's), nodes are scattered at difficult-to-reach areas and any interaction with a node after its initial setup is usually difficult. As such, applications designed for this kind of networks must be low power and ad-hoc in nature, as the topology of such networks is usually unknown. In this thesis we propose a new cross-layer communication protocol adhering to the concepts of low-power, low traffic, ad-hoc WSN's. The designed protocol utilizes *multiple* frequencies to address the link, medium access control (MAC) and routing over an ad-hoc topology, while attempting to function despite the common innate problems of such networks such as partitioning or interference.

Thesis Supervisor: Assistant Professor Aggelos Bletsas

Acknowledgments

.....

Table of Contents

List of Figures

List of Abbreviations

ACK	Acknowledgment
ADC	Analogue to Digital Converter
LBT	Listen Before Talk
MAC	Medium Access Control
MCU	Micro Processor Unit
TUC	Technical University of Crete
WSN	Wireless Sensor Network
Medium Access Control Specific Abbreviations	
B MAC	Berkeley MAC
CA	Collision Avoidance
CSMA	Carrier Sense Multiple Access
EM MAC	Dynamic Multichannel Energy Efficient MAC
L MAC	Lightweight MAC
S MAC	Sensor MAC
Y MAC	Energy Efficient Multichannel MAC

Chapter 1

Introduction

1.1 Introduction

1.1.1 Wireless Sensor Networks

In this section we will discuss the various concepts definitions that apply to a wireless sensor network.

Definition

A Wireless Sensor Network (or WSN)is a group of independent platforms, known as the nodes, carefully placed over an area, able to communicate among themselves, forming a network over which information can be circulated. Each node is equipped with a wireless radio for communication and is equipped with sensors for measuring changes to a particular set of values. Every WSN has a central node(s)that acts as the final destination for every simple node's sensor metrics.

Basic Concept

The basic idea behind a WSN is to provide a wide low cost network over which information can be transmitted and processed,even at remote locations where power supply is limited. This is achieved as each node is an independent,low cost and low power consuming platform programmed to accurately communicate with the other nodes, so as to channel the data mined from it's sensors to the network's sink or sinks.

Usual Applications

The most common application of a WSN is the surveillance of an area and the measurement of a broad category of physical values such as(but not limited to): temperature, luminescence, moisture, movement etc.

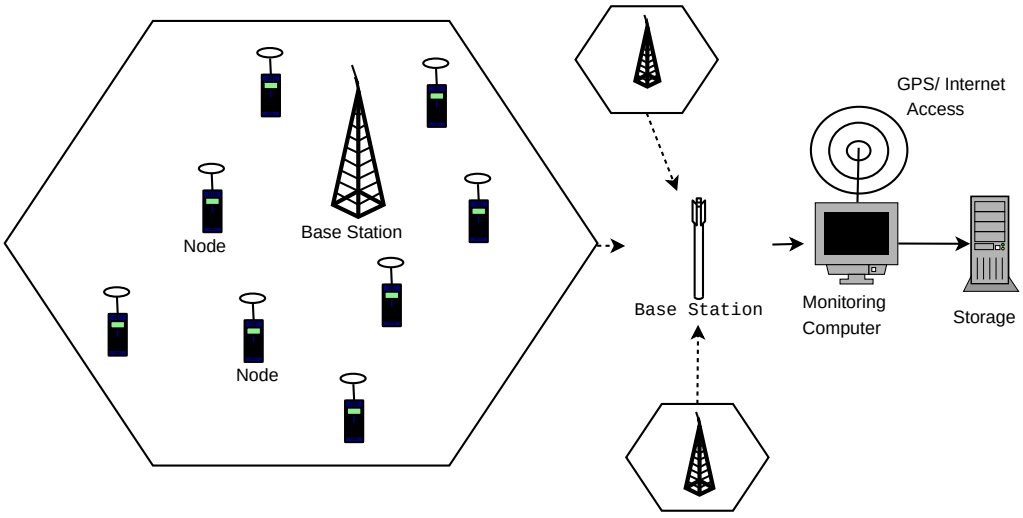

Figure 1.1: A Wireless Sensor Network

1.2 Thesis Overview

1.2.1 Purpose

The motivation behind the conception and implementation of this thesis, was the need to create a reliable robust and simple MAC that is resistant to interference and could avoid the congestion of the CSMA/CA MAC; while being easy to implement on platforms lacking an operating system.

1.2.2 Thesis Summary

Based on the aforementioned guidelines, it was concluded that the best way to complete the required goals was to utilize the benefits of the frequency domain. By using multiple orthogonal frequencies we can simultaneously

address the interference and the congestion problems. However to be able to create a useful application, a simple lightweight Routing protocol had to be implemented, so the data on the network could be circulated without previous knowledge of the network's set-up.

Finally, an application scenario had to be designed and implemented to serve a particular purpose; in this case: A smart agricultural field simulation.

1.2.3 A first approach of the individual parts presented in this Thesis

We will present a brief view into the goals and elementary design considerations of the Thesis communication protocol which will be henceforth presented into categories, the MAC layer, the Routing layer and the Application layer protocol which was largely simulated.

The *MAC* layer of the communication protocol was designed with specific goals in mind, in tandem with WSN requirements. We required Simplicity, Low Power Consumption, Interference and Traffic congestion resistance. Of course each one of these goals brought along some considerable restrictions: The need for Simplicity brought lack of global time synchronization and wake up prediction, requiring Low Power consumption meant that we need to turn on the radio and wake up as rarely as possible and being able to work around congestion and Interference mean that we had to use various metrics and mechanism so as each node can predict and counter these problems as they arise.

With the above goals and the restrictions established, we needed to implement an Extensive Collision avoidance Algorithm to prevent any unnecessary packet loss, we assigned multiple frequency channels to the network's nodes and base stations trying to grant Frequency agility to our protocol, which was achieved by performing *Channel Interference Estimation and Archiving*, *Congested Channel Avoidance*, *Base station Periodic Channel scanning*. Each node is required to keep metrics of traffic and interference for any channel visited. These metrics are update upon every new visit to the corresponding channel and are also gradually decreased over time, to bestow more

agility to the protocol; a more thorough presentation will follow in chapter 3.

The *Routing*, sharing the key characteristics of Low Power and Cognitiveness with the MAC layer was designed to be Sink oriented and featuring Multi-hop agility so as to ensure communication continues despite any unfortunate events during the networks function.

As per the MAC layer to implement the aforementioned features of low consumption and cognitiveness we had to keep the message exchange as low as possible and make any routing decisions with locally available data. With the above goals and the restrictions established, we designed an extensive indexing mechanism, similar to a cache memory's, for the required routing tables to conserve power spent on editing entries and we established austere criteria for any node candidate to act as a relay. We used a similar to the MAC layer's cognitive approach on traffic as to how to update each relay's metric and attributes so we keep traffic to a minimum. Finally, as already mentioned any decision made is done with the information a node has stored at the moment of the route planning, following a greedy approach with no need of outside coordination.

Finally, the *Application* chosen to be implemented and function over MAC and Routing layers of this thesis, was a smart monitoring environment of small garden or field.A WSN designed to serve such an environment is characterized by a) low traffic and b) Low power consumption

As such, the application needs to adhere to the following requirements:

Simplicity: Be able to run efficiently on low cost platforms with no operating system.

Low Power: As Network lifespan is of the essence, it needs to be energy efficient, which in turns means that exchanged messages must be kept at a minimum.

Low message Generation: As the application is meant for a small plant field or garden, sensed values are not subject to rapid changes, so an effective low traffic algorithm has to be implemented.

Urgent Message: The application must be able to support Urgent messages, to monitor any critical events.

Taking the stated attributes into consideration we created a simulation of a network where each node has generated packet heading for the base station, and a fraction of these packets are urgent. A simple temporal correlation based data protocol has been implemented as well, meaning that a node will not transmit a value if it has not changed more than X% from its previously transmitted value; where X is a programmable value given at the start of the simulation. With the above goals and the restrictions established, the following basic features needed to be implemented: a) Simple Quantitative criteria for the transmittable values, b) Critical value monitoring and c) Smart Sink based view of the network

1.2.4 Prior Art

There has been significant research in the communication layer of the WSN's field of study, whether is at the MAC Routing or physical layer. Hereafter follows present s brief overview of this work, categorized by the frequency utilization of each protocol.

Single Frequency MACS

S MAC [7] A MAC and Routing cross layer protocol, this is a synchronized single frequency time division protocol, that has node form neighbourhoods and assign specific time slots to each node, through messages that act as network discovery and synchronization messages.

L MAC [4] An adaptation and optimization of time Slot TDMA scheme for WSN's, this protocol assigns with a distributed algorithm Time slots to each Node of the Network, exactly one slot per Node, and aims to reduce power consumption and overhead by getting rid of the CTS / RTS scheme of the CSMA/CA. The synchronization required for the time slot architecture is performed by an exchange of control messages,

similar to those of S MAC, that also carry info on every Node's assigned slots. With the above mechanism carrier sense is deemed as unnecessary and is avoided.

B MAC [10] A protocol focusing more on the link aspect of inter node communication, B MAC splits time into slots and each slot into a number of mini slots. It separates each slots into 2 categories, data slot and contention slots. Any contention happens in the mini slots and any data exchange in the data slots. After a period of normal slots, a time slot is marked as a *broadcast* slot, where multi recipient communication occurs. B MAC uses preamble suppression to force any overhearing node at the moment of transmission to sleep and not interfere with the data exchange.

CSMA [9]: An asynchronous, protocol considered a cornerstone in MAC mechanisms, CSMA is based on human verbal communication patterns, where one can talk only if he is sure that no one else is talking at the same time.

Multiple Frequency MAC'S Some of the most important protocol in the multiple frequency category of MAC's:

ML MAC [8]: An improvement of The L MAC, ML MAC uses the foundation of L MAC, but extends its functionality by using multiple frequencies. Frequency assignment data is encapsulated to the control messages, so now every node has a specific frequency as well as a specific time slot. By using multiple frequencies ML MAC is somewhat resistant to interference and can accommodate more nodes with grater ease.

Y MAC [6]: This synchronous protocol employs both TDMA and FDMA strategies to ensure communication. It uses the same time frame architecture as B MAC, where time slots are further divided into mini slots and has periodic broadcast slots. It separates frequency channel into control and data channels, where handshaking for a node's or sink's

attention occurs on the control channel and further data exchange on one of the data channels. With this way the authors ripple out traffic into multiple frequencies reducing congestion. Any broadcasting occurs into the control channel. If the control channel is overloaded with interference Y MAC nodes will hop to the next channel according to a channel hopping mechanism; usually a linear one (for channel 1, hop to channel 2).

EM MAC [3] This synchronous protocol requires accurate real time clocks and provides great frequency agility and interference avoidance by enabling nodes to dynamically hop to new channels according to how reliable these new channels are. Reliability is performed by assessing the traffic and interference of each channel. EM MAC allocates a specific time slot to each node where any communication either towards or from that node occurs. it employs a receiver based strategy, where potential sender must wait for a special beacon packet a node transmits when it wakes up, just before being ready to receive. At each time slot, before waking up or sending the ready beacon, will hop to a new channel according to some hopping mechanism. With this strategy a node always hopes to new channel at different times, and every sender wishing to communicate with said node, has to accurately calculate the recipients next frequency channel and its exact wake up time.

The following section confers a thorough analysis Of the Multiple Access Control protocol designed and implemented in this Thesis.

1.3 Definition

A **Multiple Access Control** (hence: MAC) protocol is an algorithm designed to guarantee the reliable communication between multiple platforms sharing a common communication medium. Every MAC protocol is characterized by its Packet Delay, its packet Error Rate, its Delivery Ratio, Throughput and its Power Consumption. The Packet Delay is the mean time passed between the particular packet's generation and its delivery to

the intended recipient, the *Packet Error Rate* is the mean percentage of the packets that failed to reach their intended recipient or were delivered with error (due to interference caused by the environment or other nodes' simultaneous attempts for communication) whereas Packet Delivery Ratio is the number of Packets successfully delivered / Total Packets Sent. Throughput is the packets successfully delivered to their intended recipient divided by the total packets generated in the network, calculated at a given time interval and finally the Power Consumption of a protocol is the mean Power consumption for its regular operation.

1.4 Categories

In general there are many ways by which we can categorize MAC protocols, the most prominent are to separate MAC's according to their Time organization approach, which can be a Time frame scheme Architecture, also known as synchronous protocols or with no time division approach, known as asynchronous protocols. Another way is to categorize a protocol based on the frequencies utilized for communication; single or multi-frequency protocols. Given that main focus of the Thesis was to design a multi-frequency communication protocol we will present a brief overview of the most important aspects of Single and Multi-frequency schemes.

1.4.1 Single Frequency MAC'S

The purpose of the protocols falling into this category is to provide reliable communication between 2 or ore participants who share the same medium both in time and in frequency domains. By using a single frequency we gain the benefit that the entirety of the Network's Nodes are in the same channel; thus conferring some significant strengths and weaknesses, as described below. The main advantages of the MAC's falling in this category are the narrow spectrum required for their operation (due to a single carrier frequency), the ease with which broadcasting and synchronization can be be supported and usually a relative simplicity of the various mechanism re-

quired, at least when compared to multiple frequency handling. There are also a number of disadvantages for the MAC's falling in this category most notably the innate *interference weakness* as the single carrier frequency used for communication makes these kind of protocols vulnerable to environmental or intended interference, which can drastically degrade their performance or even complete block it. Another major problem is congestion due to*density*, as the single carrier frequency used means that the entirety of the networks Nodes will share one channel, meaning that at any given time if within transmittance range, every node can overhear the communication of all the other nodes. This, in dense networks, means that communication can quickly become congested due to multiple collisions of the various senders and delay from the backing off mechanisms that try to avoid the said collisions. Finally, the infamous *hidden terminal* problem which can be summarized as follows: a node just entering the network or having been away for an extended period of time, does not have an accurate estimation of the current traffic, so it can disrupt an ongoing exchange (say between A,B) in the following way: Having missed the handshaking of the communing nodes, the node now entering the channel falsely deems the channel safe for transmission, as the rest of the nodes within transmission range have backed off to allow nodes A, B to complete their exchange. Thus, the Node will transmit a packet in hope of reaching the intended recipient; possibly causing a collision.

1.4.2 Multiple Frequency MAC'S

Like their single frequency counter parts Multiple Frequency utilizing MAC's provide a means of reliable communication. The ability to use a wide frequency spectrum gives them an additional powerful tools in their disposal: the ability to organize the Network's Node both in time and in frequency, by assign different frequency channel for communication and possibly also time slots. The main purpose of the protocols falling into the multiple frequency category, is to utilize a number of orthogonal carrier frequencies to achieve simultaneous, interference free communication among Nodes within each others transmission range; thus possibly decreasing the Delay and con-

gestion of the network. Thisbestows them with the significant advantages of *Interference Resistance*, *Congestion Resistance* and *Increased throughput* under some circumstances.

The main disadvantages of the Multiple Frequency schemes are potentially *increased Delay* due to the possible required handshaking and frequency hop coordination over multiple channels, which could also bring added *overhead* and perhaps *increased power consumption.* It is imperative for one to remember that the above are just indicative of the main problems one has to be alert for when designing or evaluating a Multiple Frequency MAC protocol; not a native problem for each one.

1.5 Problem Statement

1.5.1 Main Goal:

The main goal and purpose of this Thesis was to design *and* implement a new Multiple Access Control protocol specifically for Wireless Sensor Networks consisted of Nodes functioning with no operating system.More specifically the new MAC had to be:

1. Multiple Frequency utilizing
2. Low Power Consuming
3. Able to operate on low cost platforms.
4. Distributed
5. Asynchronous
6. Configurable
7. Expandable

The reason the aforementioned goals where set, is that the MAC was designed mainly to be used my small remote independent WSN's monitoring a small area with relatively low traffic (i.e. a digital garden or a home monitoring

environment).With this direction in mind it will now be explained thoroughly *why* the designed MAC had to be ruled by the stated constraints.

1.Multiple Frequency: First and foremost was the ability to be able to communicate despite any interference. Interference resistance was imperative because in small private or home environments there is a usually a large number of electronic devices continuously operating; transmitting intentionally or simply radiating unwanted noise possibly due to a damaged exterior or poor design.
It was thus deemed necessary that the new MAC should be able to handle such interference and be able to function, even with diminished throughput, despite any noise.

2.Low Power Consuming: A secondary but also a very important constraint was to be as low power as possible (keeping in mind that the algorithm has to asynchronous).This is extremely important as most WSN's have a number of nodes that are difficult to reach once they have died out, and their recharge is a tedious task best performed as rarely as possible.

3.Low cost platform compatibility: Another significant constraint, requiring no state of the art hardware or any kind of existing operating system.The MAC algorithm had to be able to function in a platform equipped with an MCU with an architecture that supports C.

4.Distributed: The algorithm had to be completely distributed with no central coordination whatsoever.This was required because it was needed to able to function in a multihop environment.

5.Asynchronous: The absence of a built in system clock in low cost platforms and the difficulty in implementing an accurate one lead to the requirement that the new MAC had to function asynchronously.

6.Configurable: All its metrics and variables should be highly tunable so as to able to support as wide a set of applications as possible.

7.Expandable: The protocol had to be designed in a modular architecture, so that each of its aspects can be easily expanded or improved, with out the need of complete structure overhaul.

Whether one designs a telecommunications test bed or a wireless application, the need for reliable communication between the nodes that make up the network, is the same. Either for financial or educational reason many WSN's are made using custom made nodes that lack some of the high end feature offered by the high end commercial products. So, the need of a reliable and lightweight,in terms of processing power and memory resources,MAC protocol able to operating on such platforms is apparent.
But *why* a new MAC protocol? Because it was imperative to have a MAC protocol suited to the needs of the specified low traffic situations, and to the best of our knowledge no other was found that offered such simplicity in implementation while simultaneously offering the benefits of channel hopping and interference resistance and recovery. Also the accumulation of knowledge on the basics of WSN MAC protocol design, fine tuning and testing was a significant factor. We will proceed to to offer a detailed description of the designed communication Protocol, dividing it into 2 parts according to the network layer that each part refers to; *the MAC and the Routing layer*.

Chapter 2

Medium Access Control

In This section we will view and analyse the mechanics of the designed MAC protocol.

2.1 Overview

The basic idea of the algorithm was derived from the CSMA protocol, where each node listens for a period to the channel, and if found free, transmits its packet; if it collides it usually employs the functionality of some Back off mechanism to solve and further avoid collisions.CSMA works quite well in low traffic situations (such as those described until now) but has a major weakness: it is vulnerable to interference, and traffic congestion.So as we wanted to avoid the synchronization overhead and complexity that comes in a very low traffic network we came up we the following notion: Adjust the CSMA protocol to utilize multiple frequencies. Again adhering to the constrain of low power and complexity, each node is assigned to a specific frequency, essentially forming frequency clusters. Similarly a sink is assigned its one channels one for control and single packet messages and one for multiple packet messages (burst transmission). So, a network is distributed into virtual clusters, where each cluster has the nodes that share the same frequency. It is possible to switch to a new channel and reside there if a pre set channel is deemed as congested or interference heavy. So in *summary*, this protocol is a modified CSMA procedure spread out to multiple frequencies, with mechanisms to ensure one node can locate another despite any interference and initiate a Listen-Before-Talk exchange, improved by a selection of features that will be described below.

2.2 Architecture:

This section describes the Architecture of the protocol.

Network Architecture: Each node in the network can fall into 2 categories:

Base Station: A node acting as a base station is always active,i.a considered to have abundant energy and the majority of the information circulated in the network is meant for this type of node.It resided on a pre set number of channel as it will be described later on.

General Node: A node acting as a general node is essentially a typical node, the workhorse of the network that is expected to be supplied with some sensors, executing a data protocol and has a finite amount of energy.

MAC Frequency Architecture: Each node of the 2 aforementioned categories has a pre allocated set of channel frequencies where it resides, performing its various operations. These channels are decided and distributed during the set up of the network according to the given available frequency spectrum and the distribution Algorithm. The algorithm chosen in this thesis is hash like function which allocates frequencies to nodes based on their Id number and the available amount of usable frequencies (which is a given set). After a node is assigned a channel, and given that the distributing function is known for each node, every node in the network can calculate any other node's residing frequency in the network with no additional overhead. The assigned channel of a node is hereafter called as the native channel.

Interference Detection: Every Node in the network can assess a given channel for interference levels.We have designed a metric to monitor this interference named *channel badness*. If the node detects unusual traffic, such as non interpretable packets or packets with unknown format, then it will activate the channel assessment algorithm trying to evaluate the channel that

this anomaly occurred.If this anomaly occurs often enough to disrupt reliable communication then the channel is deemed as Overloaded with interference and the node invokes the Channel hop mechanism which is described below. It is possible to channel hop without finding the channel overloaded however; if the channel has unusual traffic that is not frequent enough to completely disrupt communication but may cause problems, the channel is branded as having some interference but not a debilitating one. The assessment algorithm will increase the badness metric by a variable value and if the new badness is high enough to be considered as bad as an overloaded channel, then the node will channel hop.

Channel Hopping: When required, each node can invoke the *Channel Hop* algorithm which is responsible for switching channels. There are many algorithms one can employ for this task, the one used in this thesis is a sequential, linear algorithm which assigns the next in line channel; if that channel's *badness* qualifies as usable.

It is important to note at this point, that any time a node is forced to leave its pre set channel, it will set an interval at the end of which it will try to re-assess its *native* (pre set) channel; if found free it will return, else-wise it shall remain at its new channel.

If it the new channel is assessed as unusable the node will hope to the next in line channel, assess it, and if found clear stay there, else continue the process until a free channel is found. If no free channel can be found, the node will sleep and try again at a later time. Any other channel than the pre set channel of each node is hereafter called the *residing channel*.

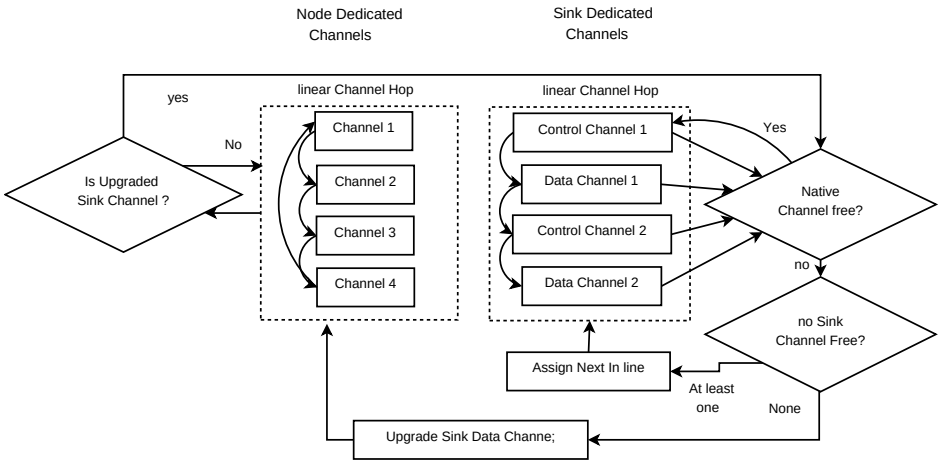

Figure 2.1: Channel hopping Sequence Flowchart

Locating a node: When a node needs to locate another node in order to perform an operation (i.e to transmit a packet), it must calculate the current residing channel of the target node. Given a distributing function that is known to all node in the network, the searching node will first attempt to reach the target in its pre set channel. If that channel is deemed as *bad* then the would be transmitter will invoke the channel hop algorithm and try to locate the node at its other residing frequencies. If the node is not located and a fixed amount of time-out intervals occur, then the node will try at the pre set residing channel of the target.It will assess the channel and if found clear of interference it will search for its target there.If it still cant find the node, it will sleep for a given interval and try again later; provided the operation to be performed is not urgent.If it is marked as urgent then the node will follow the above steps until it locates the node or until a pre set threshold of tries is reached, at which point it will mark the target node as cut off from the network.
If the node locates its target either at the very start of its communication or at some later point, it will perform the intended operation and then return to its residing channel and sleep.

If it is unable to reach its target but the new channel is free of interference, the node evaluates 2 possible scenarios:

1. Time out die to some random fact such as bad synchronization
2. Target node hopped to another channel.

To address these 2 scenarios, every node keeps info of each available channel's badness and traffic metrics. If the amount of time-outs as a given channel exceed a preset number the transmitting node will assume that the recipient of its message has hopped to the next in line channel.

Contention: When a transmitting node selects a channel that is not overloaded with interference and the recipient node is believed to reside at, it till initiate the contention algorithm to avoid unnecessary packet loss due to collisions.

The transmitting node, hereafter the sender, will listen to the medium for a given interval (currently chosen as one packet transmission time) and if the channel is found free, it will transmit its packet; a mechanism known as Listen-Before-Talk. If the channel is found occupied at the time of the probing before the transmission, the sender will back off and retry.

Each time the medium is found occupied the sender increases a counter (*traffic indicator*), which uses it to keep track of the traffic in the current channel. Based on the value of the traffic indicator, the sender will back off for increasingly greater periods until the traffic indicator surpasses a configurable value (*low traffic threshold*); at which point it will use probabilistic probing after each back off period. The sender will now listen to the medium based on the value of a random variable uniformly distributed at [0 : Times channel found occupied], if the value or the random variable is less than 1, then the sender will once again initiate Listen Before Talk, else it will sleep and re-check the value of the random variable when it wakes up again. As this probabilistic listen before talk could lead to extended periods of contention at a traffic heavy channel one can employ the utility of various mechanisms,like the Suppression feature explained at the Fairness section below, to resolve these issues.

```
Pole the Channel;
if clear then
    transmit;
else
    while Channel NOT clear do
        Back off;
        Increase traffic indicator;
        if traffic indicator > low traffic then
            probabilistic listen before talk;
            if Generated Random variable x < 1 then
                listen to channel;
            else
                Sleep;
            end
            if Times backed off > Unfair times then
                employ a Fairness feature;
            end
        end
    end
end
```

Algorithm 1: Contention Algorithm

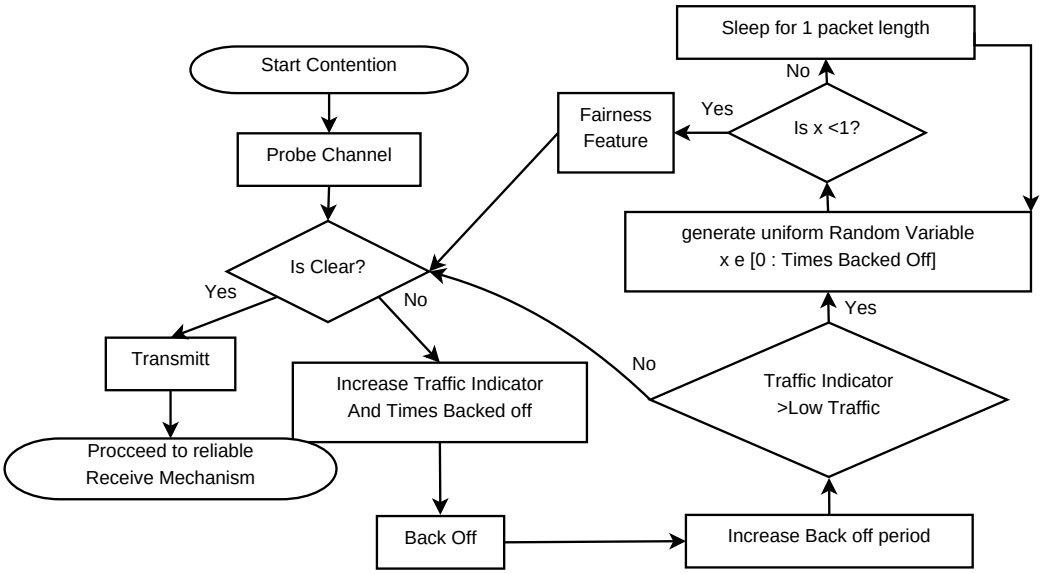

Figure 2.2: Contention Flowchart

Traffic Assessment: When ever the node collides in contention or receives a foreign message at a given channel it will increase the channel's traffic indicator, which is a metric of the speculated data traffic on this channel.When that indicator surpasses a threshold then the node activates various mechanism to safeguard itself an the network's performance from congesting traffic.Likewise each time out or successful transmission occurring on a given channel will decrease its traffic indicator.

Link establishment: After a node has wan the contention and has transmitted its packet it is now imperative to establish a reliable link. For this reason the sender has the ability, as per the normal CSMA protocol, to request an acknowledgement message from the receiver, so as to be u sure its packet was delivered correctly, In the event of a time out (No ACK message received within a configurable window) or interference during this specific interval that an ACK is been waited for, the node has a number of retransmission until it will seize its efforts to reach its target and retry at a later time. In the event of a foreign message of a another node during this ACK wait interval, the sender will increase its traffic indicator metric and retry to

send by reentering contention also decreasing the number of retransmission for the specific packet. The packet is considered to be lost as it is most likely that the ACK message has either collided with the interrupting message or arrived before the sender could reenter receive mode, so it must retransmit its packet. During the link establishment the node has the ability to employ a Fairness feature to ensure that its packet will be heard correctly, especially in high traffic channel. For example, it could force nodes within listening range of the current communication to go to sleep for a specific interval, so the do not unintentionally interrupt the active data exchange.

Modules selected: Adhering to the configurable (6) and expansiveness (7) constraints we had to select specific algorithms to integrate to the designed protocol to handle the various functionalities. Here follows a list of the selected algorithms serve each functionality:

1. Listen-before-talk: The node will probe the channel sensing when the signals on air are above a programmable threshold.If above the threshold then they can cause interference, so the node will wait and probe again at a later interval as described in Contention: 2.2 on page 26.

2. Back off: When required the node will back off by sleeping the base time required to transmit a packet linearly scaled by the times collided in the channel, or in the case of burst transmission (if a node happens to hear any packet of the burst transaction)for the length of the burst transmission plus one packet.

3. Contention resolution: Each node participating in the resolution will back off a random interval between one packet transmission time and Times collided X packet transmission time.After that interval the node will examine the value of a generated random variable x by a uniform distribution and if x¡ 1, the node will probe the channel, else it will further sleep for one packet transmission time and perform the same procedure with the random variable x.

4. Frequency allocation: Each node during network initialization will receive

a pre set channel according to the result of is Id mod the available frequencies.Henceforth, this is its native channel.

5.Channel Interference assessing: The node will power on its radio and probe the channel for programmable amount of time, preferably an extended interval capable of giving the node an accurate sample of the current traffic on the channel.The Node will count the times it finds signal above a programmable threshold (*negative*), and the times it find the channel with signal below the same threshold or completely free (*positive*). The signals above the threshold are branded as strong (capable of interfering with transmissions) and those below the threshold are marked as 'weak'. If the times found occupied by strong signals are considerably more than the times that weak signal are on air, then the channel is tagged as *Overloaded*. If strong signal are comparable to the weak signal in terms of times encountered, then the channel is tagged as *Occupied*, on any other occasion the channel is tagged as *Clear*.

6.Channel traffic assessing: The node categorizes the traffic at a given channel at 3 distinct categories and handles every increase/ decrease with a different perspective dependent on the current traffic level:

1. *Low traffic*: Traffic falling into this category symbolizes either low actual data exchange, or insufficient overview of the real traffic. The node increases the indicator by 2 for each foreign message or contention loss and decreases it by 1 for each successful transmission.
2. *Medium traffic*: Traffic falling into this category symbolizes a relatively busy channel, at which a node will increase the traffic indicator by 5 and decrease it by 6
3. *Heavy traffic*: Traffic falling into this category symbolizes a very buy channel.Care must be taken as a collision is highly probable, degrading network performance.The node increases the indicator by 1

7. Channel hopping:A linear selection has been chosen, so when a channel hop is required the node will compute the arithmetically next channel in line from the current channel. For *example*: Node 11 wants to transmit to node 15. Node 15' s native channel is channel 18.Node 11 has previously assessed channel 18 as bad, so it must invoke channel hop to find the next frequency that Node 15 could reside. It will select the arithmetically next channel which is channel: 19. If no channel satisfies the constraints of channel badness etc, then the Node will sleep and try again at later time.

   The mechanism works slightly differently when dealing with sink channels.While a Node will still try to hop to the next channel as with the node data channels, if no pre set sink channel can be found free, a node channel will be upgraded and temporarily serve as a sink channel.As explained in paragraph: 2.2 on page 24 a node will always try to return to its native channel, and this also applies to a sink, thus the temporal servitude of a node data channel as a sink channel.Below is a graphical representation of the linear algorithm.

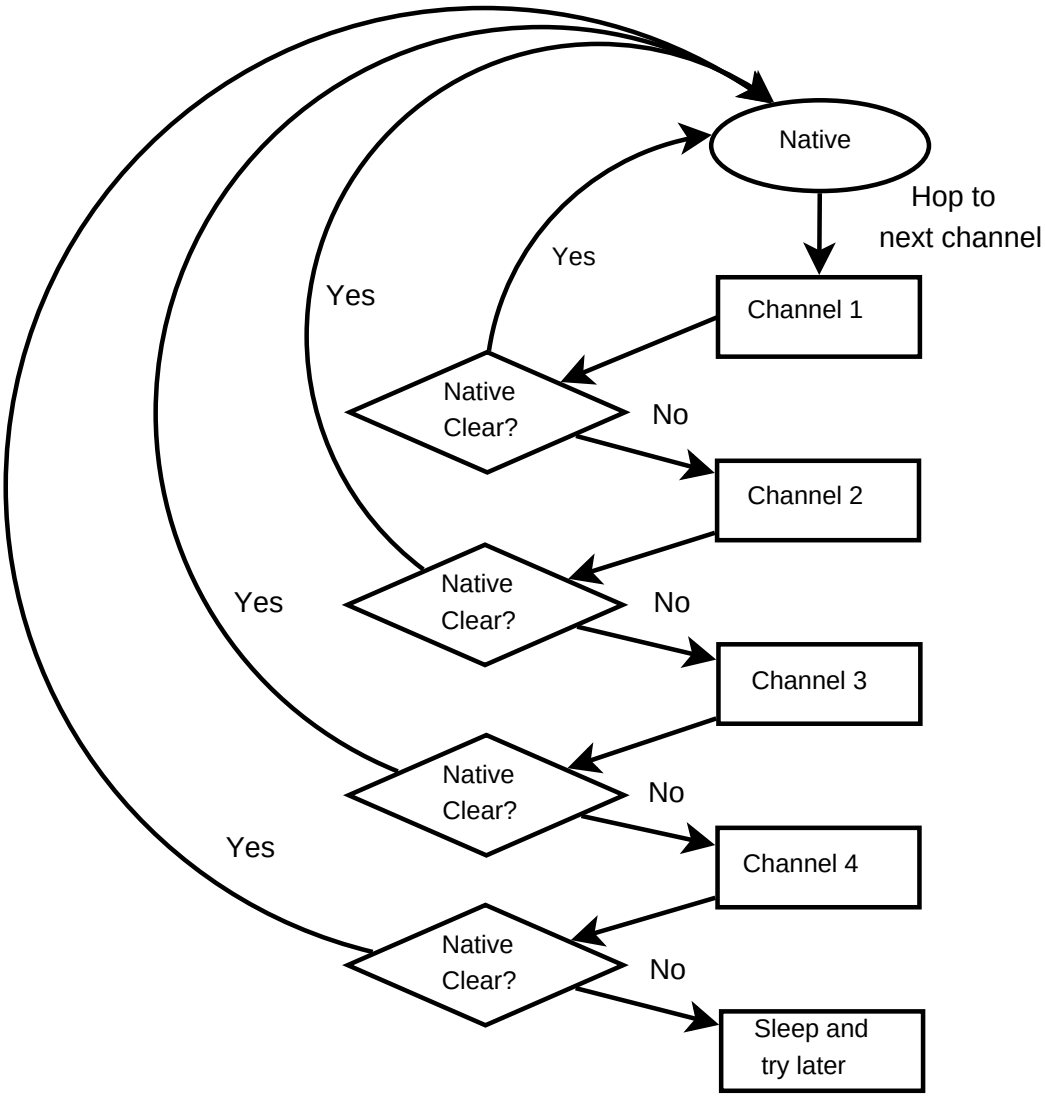

Figure 2.3: Linear Channel Hop figure

8. Fairness: To minimize node unfairness in our network we chose embed the following features into our protocol: mandatory sleep after each successful transmission, if the network is under heavy traffic and a suppression mechanism which enables nodes that have been silenced for too long during contention, to transmit a packet that forces all listeners to enter a prolonged sleep, in order to minimize collisions and thus energy consumption.

2.3 Functionality Description

Initialization: During The initialization phase each node is assigned a channel, where it hops to and assesses it to check out if it is clear or not.If it is it resided there(this channel is called the native channel of the node), if not clear it shall hop to the next in line.

Functioning on a channel After the settling phase of the Initialization period, each node will be on a frequency channel, sleeping to conserve energy, waiting for to handle any interrupt that presents itself. An interrupt can happen in the form of:

1. Message arrival meant for the node
2. Message arrival meant for a different recipient than the node,known as Overhearing
3. A sensor reading that requires to be sent by the node
4. Interference

In the event of a packet arrival meant for the node, the node wakes up receives and processes the packet and sends out an ACK if one is requested. It will then proceed to reenter sleep.

If the packet heard was a packet meant for another node, the node will enter sleep mode with its radio turned off to conserve power, for the speculated duration of the foreign packet exchange, which is the time needed for 2 packet transmission for each packet the original sender of the packet wanted to transmit. We assign the time equivalent for 2 packet transmissions to cover the time needed for an ACK answer as well, which by default is enabled. In the rare event that a node wishes no ACK, the sleep period is determined as one packet transmission time per packet length of the overheard packet.

In the event of a *sensor interrupt* the node will wake up, read the sensor value which caused the event, encapsulate the value into a packet and transmit it to the suitable recipient

In the event that the node picks up consecutive *interference* readings it

will increase the badness per reading, assess the channel if the badness is high enough and if the channel is deemed as overloaded with interference it will hop to a new residing channel as per the channel hop procedure. the phase *Functioning on a channel* is actually the steady state procedure of a node in the network where it stands ready, in a low power mode, to handle every programmed situation, invoking the transmission procedure described above in 2.2 at page: 23 and shown at figure 2.4.

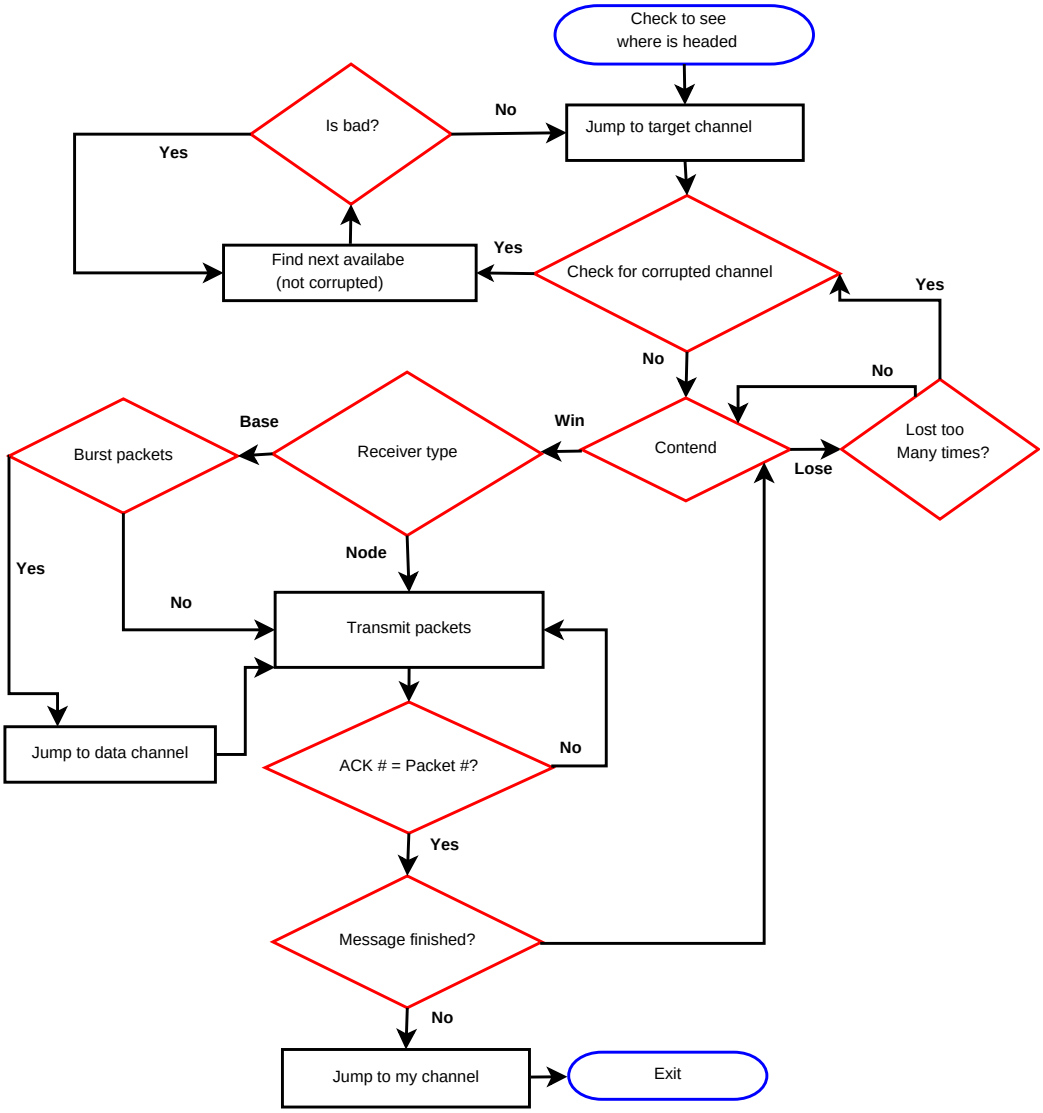

Figure 2.4: Transmit message Flowchart

Chapter 3

Routing

3.1 Definition

Routing in communications, is the layer responsible fore finding a viable and if possible,optimal, travel path for the transmittable data given that the destination is known.

3.1.1 Categories and characteristics

There are 2 basic categories in which algorithms fall according to perspective with which they examine the routing problem: Greedy and exhaustive algorithms.

Greedy Algorithms follow a greedy logic, computing and choosing the optimal choice with only local criteria. Their main goal is to provide a solution that has an acceptable efficiency, in a reasonable amount of time. They has a relatively easy implementation, a fairly fast computation time, require relatively few resources to function properly, have acceptable efficiency and require only local knowledge, or view, of an otherwise extensive problem. While the necessity of only local knowledge is a highly appealing feature, it is also a weakness as having only a near view of the problem can lead to problems or anomalies further down the path and the final solution might not be an acceptable solution,even though the partials leading to it where optimal within their scope, as its the sum of all the local solution along the path.

Exhaustive Algorithms' main goal is to function, after they acquire an extensive overview of the problem, with as many data as possible and utilize the above to converge to a catholic optimal solution. They require an accurate and thorough overview of the problem but using that extensive view,

computation converges to a truly optimal solution; leading to impressive efficiency. It is obvious though that while an optimal solution is quite sought after, having an accurate view of the whole problem at each step is not trivial at all; it requires more resources than greedy algorithms, a relatively lengthy initial set up interval and also costly and time consuming periodic updating procedures to maintain their efficiency. Adding each of these drawbacks, might cause slow computation times; a feature not sought after in WSN's.

Comparison Comparing these two broad categories, we can come to the conclusion that a selection of the two heavily relies on the problem at hand and the hardware resources available.
For a stationary TCP/IP network for example, with powerful hardware for Nodes and wired connections an exhaustive algorithms is probably the best choice, as the computation and resource requirements can be covered and mobility is not an issue.
On the other hand, in a WSN network where the medium and the network might be subject to change multiple times (either due to mobility of the nodes or an unexpected obstacle) a fast greedy algorithm might be the best choice, as it needs little updating and coordination.

3.2 Algorithm

In this Thesis we chose to design and implement a greedy algorithm for is fast computation, almost on the fly, little added overhead and most importantly for the fact that this approach does not require constant update from the majority of the network or any central coordination.

3.2.1 Overview

In WSN's the lack of wired connections means that one must find some way of gauging the quality a link based on the incoming signal (from the receivers point of view.) There are two main ways to do this:

1. *Absolute distance*:If we encapsulate a time stamp on an outbound packet so the receiver of the message can compare the departure time of the message and the reception time and calculate the time of flight; then derive the distance separating the two points based on the propagation speed of the packet. This particular method required a real time, globally synchronized watch in each node.

2. *Signal Strength*:Another way would be to measure the signal strength of the received signal and create e relative distance based on the link strength.
   This method does not require a watch, but cannot produce absolute results. The perceived strength in relative as a great many deal of factors can influence this metric.

Given that the designed Mac is an non synchronous protocol the absolute distance method was non applicable. So we chose to evaluate a link quality based on the perceived signal's strength.

Basic Function Given that each node can gauze the strength of any valid incoming signal, it can categorize the transmitter as being near or far based on the link quality.
SO every node upon hearing a packet,irrelevant of destination, it will compute its strength and invoke an algorithm that compares the link quality with any other link similarly recorded and categorizes accordingly. With this procedure each node has a table of link sorted by their relative strength and can choose on the fly which link to choose to forward its own packets. This Routing protocol also takes into consideration the remaining power of a Neighbour. If its power falls too low, a node will try to forward its packets from an alternate relay. So, have a mixed of link quality *and* power level based data circulation.

3.2.2 Architecture

Description

The designed Routing protocol is a sink oriented protocol, that resembles a hop count type procedure in the sense that that it categorizes each link,and therefore the nodes forming the link, to relative distance neighbourhoods updating and moving a given node if need be, to a new distance category. By mentioning sink oriented we mean that all the information stored and all the optimizations performed are done so as to support the data flow towards a base station. Data flow towards node is not the goal of this routing, so each node can only route info to its immediate neighbourhoods.

Routing-MAC cooperation: This routing is designed to be able to work with any given MAC given that the Routing layer can encapsulate some additional info into the outbound messages. the new fields being introduced to a packet by this routing protocol are:

Sink access Indicator This field indicates the relative distance of the node from the sink, as perceived by the node.

Power Level This filed indicates the remaining power power of the node.It is calculated by an independent algorithm.

Sink signal Strength: Absolute signal strength between the node and the sink.Is in dBm

Routing Message Indicator: Indicates whether the message contains of requests Routing info.

There is also a **Link Access Indicator** and a **Link Signal Strength** metric, which relate to to particular link and are not encapsulated to outgoing packets; they are sued by a node to categorize its neighbour as described below.

3.2.3 Routing Tables

Each node keep a number of tables to perform the categorizing task, where it sorts and keep its perceived neighbors. It is important to note that a link with a node will be evaluated and stored regardless of the native frequency of that node. Namely the tables are:

Optimal Relays: In this struct a node keep the neighbour neighbours (according to the programmed size of the struct) that after evaluation are deemed as the optimal choice to forward a packet.neighbours kept in this table have the best link with the Node, the best link with the Base station and also the highest level of remaining power.This table is updated constantly in order to achieve a high level of reliability.

Best Relays: in this struct a node will keep any neighbour that failed to enter the optimal relays table but is still a good choice compared to the remaining nodes.In the case that something happens and the optimal relays become unreachable, a node will select one of these neighbours to forward its packets.

Ordinary neighbours: In this struct a Node will keep all its neighbours regardless of link quality.If a new neighbour is detected, after evaluation is performed to check if it is viable to be marked as either an optimal or best relay, it will be registered in this table. So this struct keeps all the neighbours of a node.

The table hierarchy is perhaps better displayed by the figure below, when we have an Optimal relays table with max size of 1 entry, a best relays table with 2 entries max size and ordinary neighbour table with abundant storing size. A new neighbour will be evaluated by the displayed procedure in the following figure:

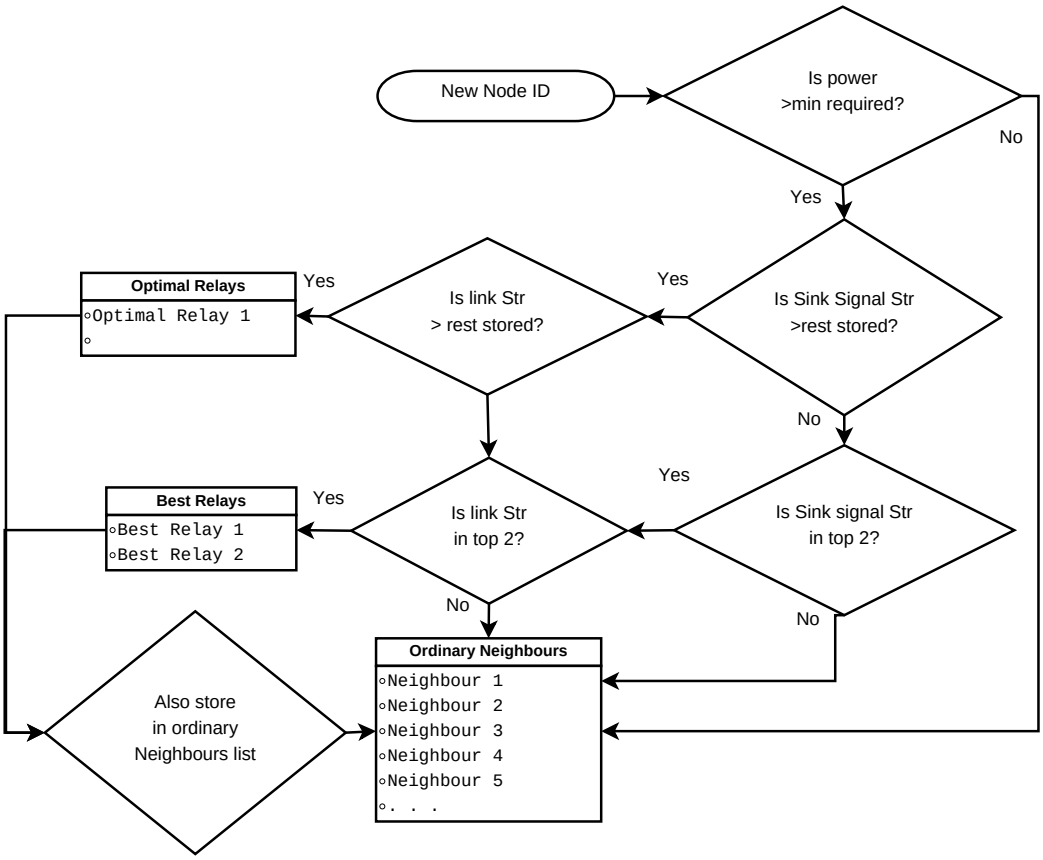

Figure 3.1: Handling of new routing Table entry

Link Partition Detection: Even if a sink is theoretically an optimal solution, there are many unexpected factors that can cause its quality to drop, such as a newly placed obstacle along the line path blocking communication. In an attempt to preserve continuity of communication in the network despite some partitioning,a detection mechanism has been implemented. Each time one of the following occurs at a selected relay's link: a) Time out b) Interference c) Consecutive foreign Interrupting messages, then the node will increase a metric called *Relay badness* which measures the reliability of an otherwise stable link. If a relay's badness surpasses a programmable threshold then the link is considered as unstable and not chosen in future attempts. In the case that the 'bad' neighbour is classified as an optimal relay, then the node will gradually decrease this badness in an attempt to check if the

instability is temporal. If it continues to be non reliable then the node will select the next best neighbour to serve as an optimal relay (which usually is the default gateway for the node's traffic). Using the relay badness notion, a node can also evaluate itself as packet relay, if its metric of Sink signal strength is superior to all of its registered neighbours.

Network Partition Detection: If at any given time, *all* of the optimal and best relays and also the node itself and all its neighbours are marked as either bad, or have no access to the sink, the node is considered to have been cut off from the network and is a Network partitioning has occurred. The routing protocol equips the node with a mechanism to combat this unfortunate situation and that is the *Network rediscovery* feature.

The network rediscovery feature has two different modes: a) Partial b) Thorough. When a node first suspects *Network partitioning*, it will initiate a partial rediscovery in which it will broadcast a routing info request message, which also contains its own info and wait for an answer; if none arrives it will sleep and try again later. In the Case that the MAC designed in this thesis is used, the node will broadcast a packet in the sink's pre set channels. at the optimal relay's channel and at its own native frequency. If no answer arrives then it will sleep and reawaken to retry. After the second attempt if no new neighbour has answered, the node will enter the thorough network rediscovery where it will start hopping and broadcasting to all the available frequencies. If even this attempt yields no results, the node will sleep for an extended period and try again at the end of that period, first performing a partial rediscovery then a thorough one.

Packet Forwarding

At any time a node needs to forward a message towards the sink it will follow the procedure below:

1) Check if neighbour entry exists in optimal relays struct;
if *yes* **then**
  Go to⟶ step 4;
else
  Go to⟶ step 2;
end
2) Check if neighbour entry exists in best Sink Relays struct;
if *yes* **then**
  Go t*o* ⟶ step 4;
else
  Go to⟶ step 3;
end
3) Update Optimal and Best Sink Relays structs by evaluating each stored Node's metrics.;
Go to ⟶ step 2;
4) Compare the candidate relay-neighbour with own metrics;
if *If chosen relay has superior link* **then**
  Forward packet through relay;
  Go to⟶ step 5;
else
  Forward packet through self;
  Go to⟶ step 5;
end
5) Transmit the message using the available MAC;

Algorithm 2: Packet Forwarding Decision Algorithm

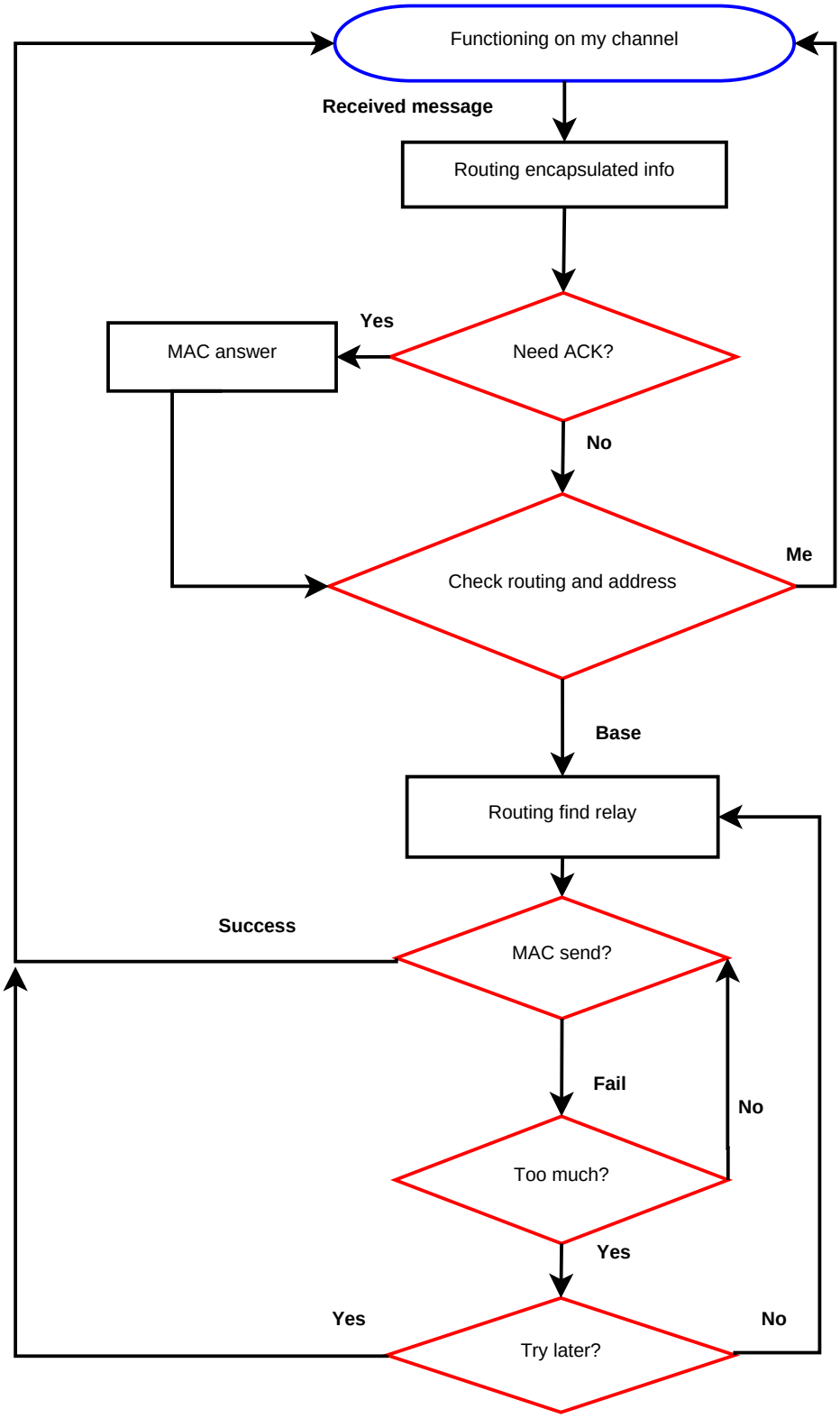

Figure 3.2: Block diagram of message forwarding

3.3 Functionality Analysis

Having presented the architecture and mechanisms a node has at its disposal to map the surrounding network, we will now proceed to present how the Routing will function in a real WSN. Its operation can be split into 3 distinct parts:

1. Initialization
2. Standard Function
3. Network Rediscovery
4. Hibernate

Initialization During Initialization every node will tune at the main sink control frequency which is usually the first frequency channel assigned to our network, (or the normal frequency of operation if any other MAC protocol than the one Designed in this Thesis is used). and start broadcasting messages with the Routing Request field enabled. With such a way every node will listen to all other nodes transmission range and categorize the newly found neighbours according to their link strength. The node will proceed and store them in the routing tables using their relative distance a reference their vicinity. If the Base station (sink) happens to pick up an message at all it will also broadcast a packet with its information so all the nodes within range estimate their distance from their sink using the received signal strength of the Base station's packet transmission. When a receives a packet from a sink during this phase it will no longer request Routing info from the sink, to reduce unnecessary traffic; it will request Routing info from ordinary nodes however.

Any node that happens to be outside the base station's transmission range, will consider the said base station to be unreachable, and forward any packet it has meant for the sink through its neighbours. In the unlikely event that a node does not register any neighbours and has no access to the sink, it will enter the network rediscovery phase which will be described later on.

Standard Function After the Initialization period is finished, the majority of the nodes have an able amount of neighbours and the network is considered stable. Each node that needs to report with the sink will use the packet forwarding mechanism described at page 43.

Network Rediscovery In the event a Node has no reliable neighbours left due to network partitioning, or missed out the initialization phase completely, it will invoke the network rediscovery mechanism described at page 42.

Hibernate In this phase a Node will enter a prolonged sleep with or with out its radio, depending on its remaining Power. During this phase a node tries to conserve power while waiting in the hope that upon wake up, the cause that forced it to hibernate will have passed (usually a network partitioning).

3.3.1 Operation Example

We will now present an example of operation in a graphical manner.

Description The below scenario depicts a simple WSN with its nodes and a base station that we will see its operation from the perspective of a specific participant: Node A. The rest Nodes are assumed to not be able to communicate among themselves for simplicity's sake, so we can describe the basic steps of the routing algorithm. At each step, we will show a graphical representation of the speculated network followed by an instance of its stored routing tables.

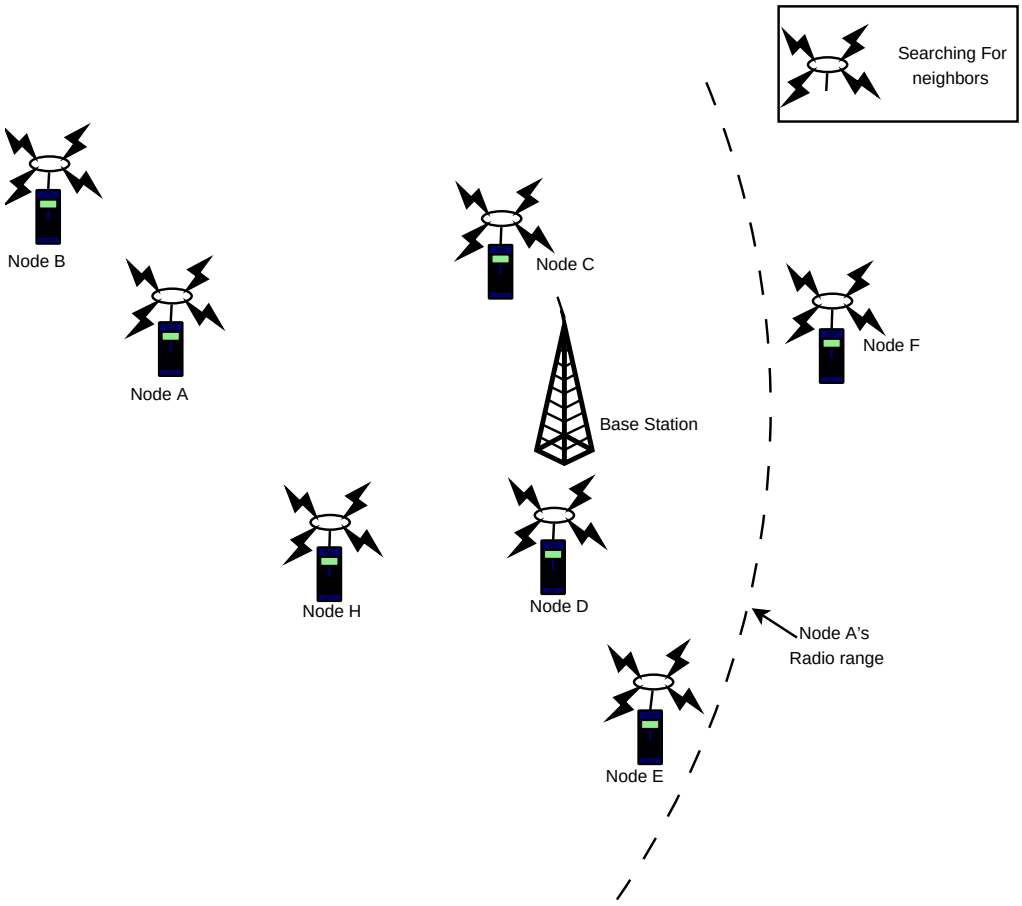

Figure 3.3: Network Initialization phase

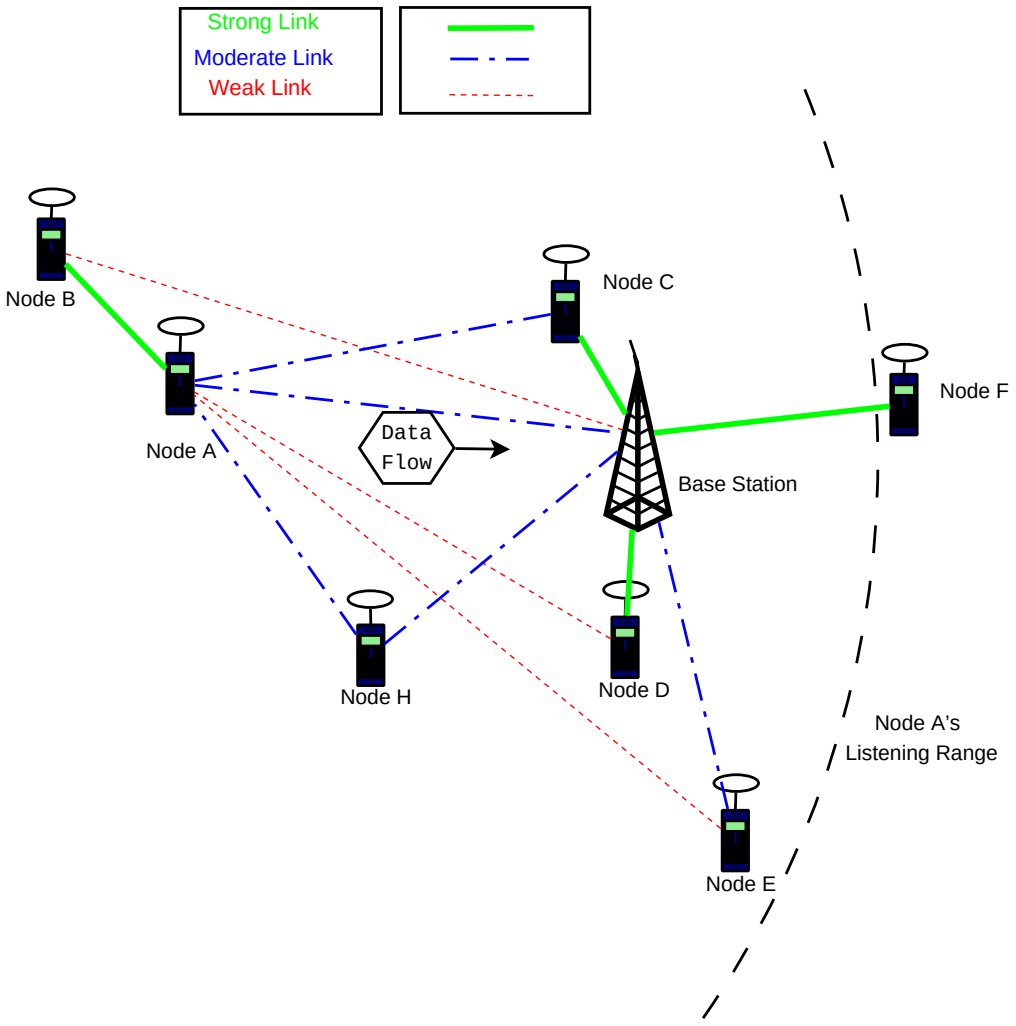

Figure 3.4: Network After Initialization phase

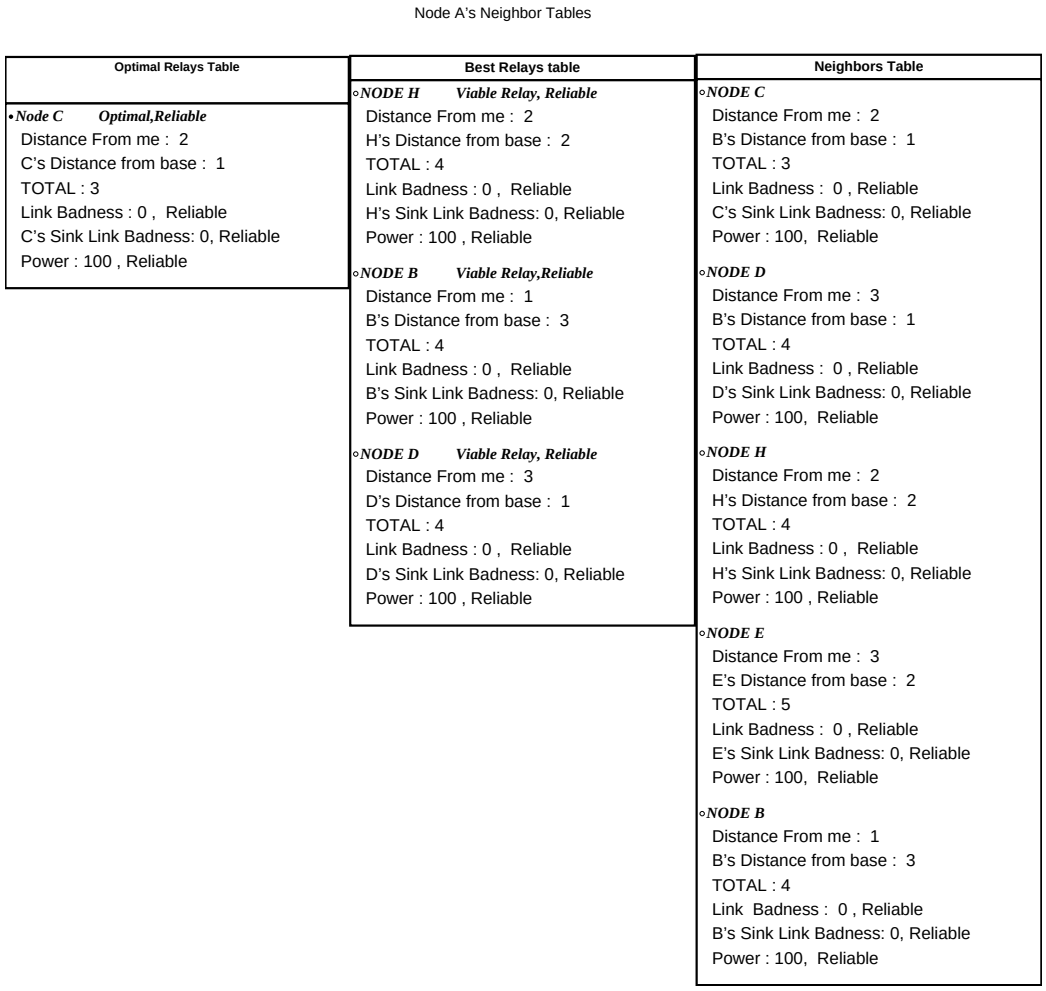

Figure 3.5: Routing Tables After Initialization phase

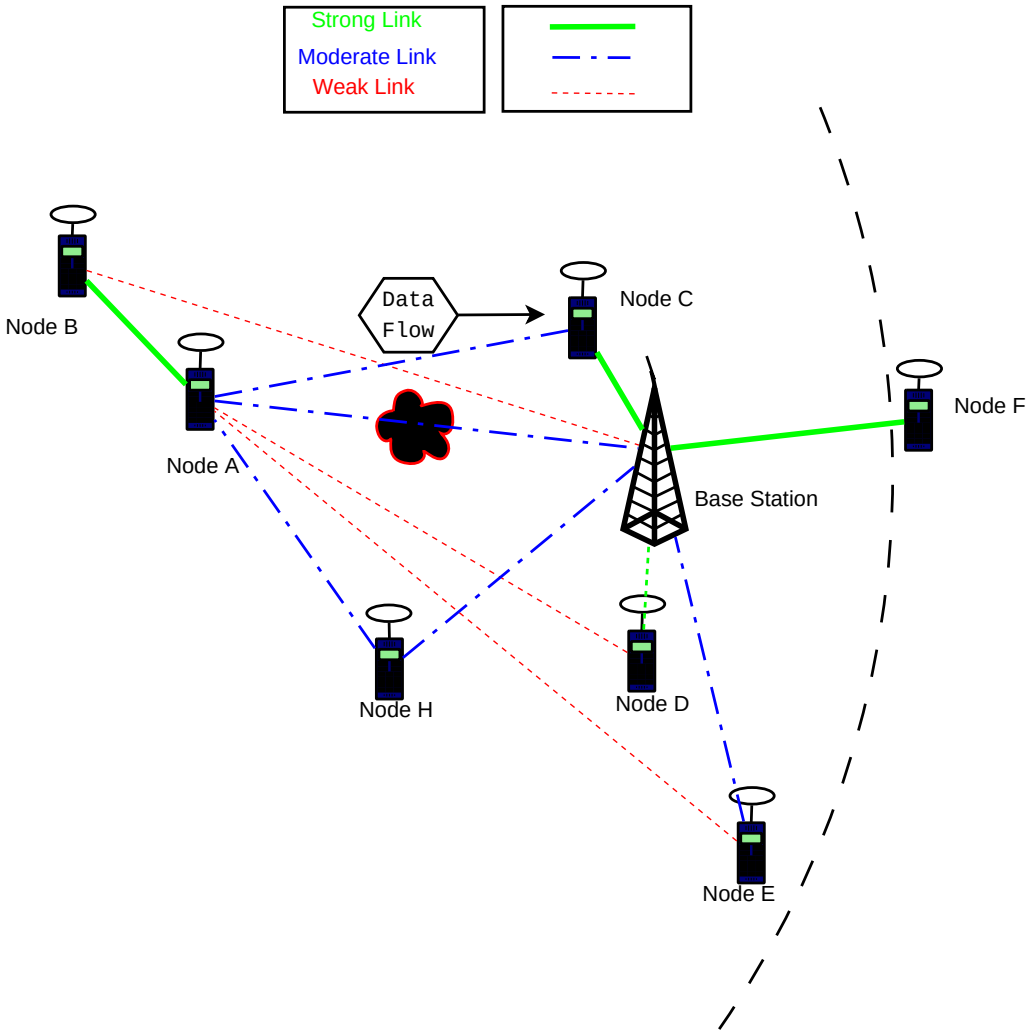

Figure 3.6: Network After Obstacle blocks A ⟶ Base link

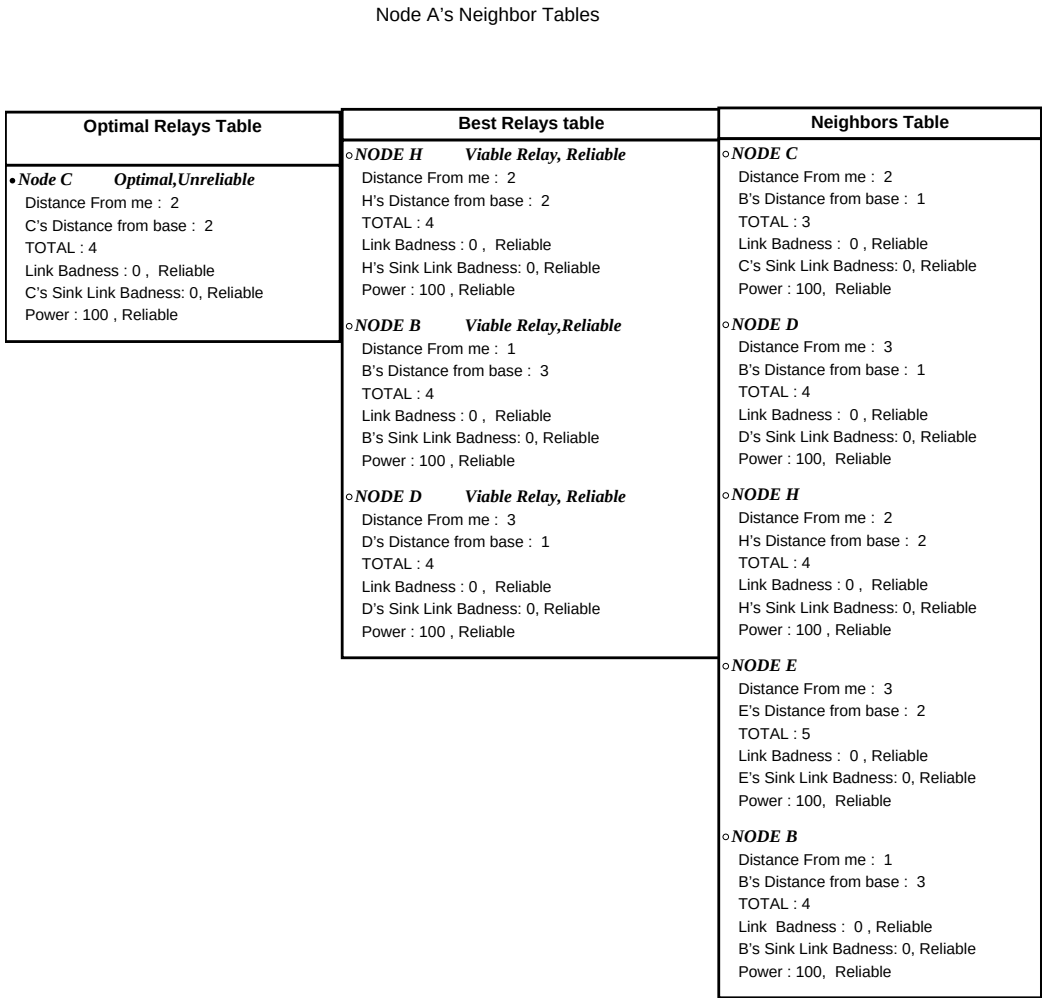

Figure 3.7: Routing Tables After Obstacle blocks A $\longrightarrow$ Base link

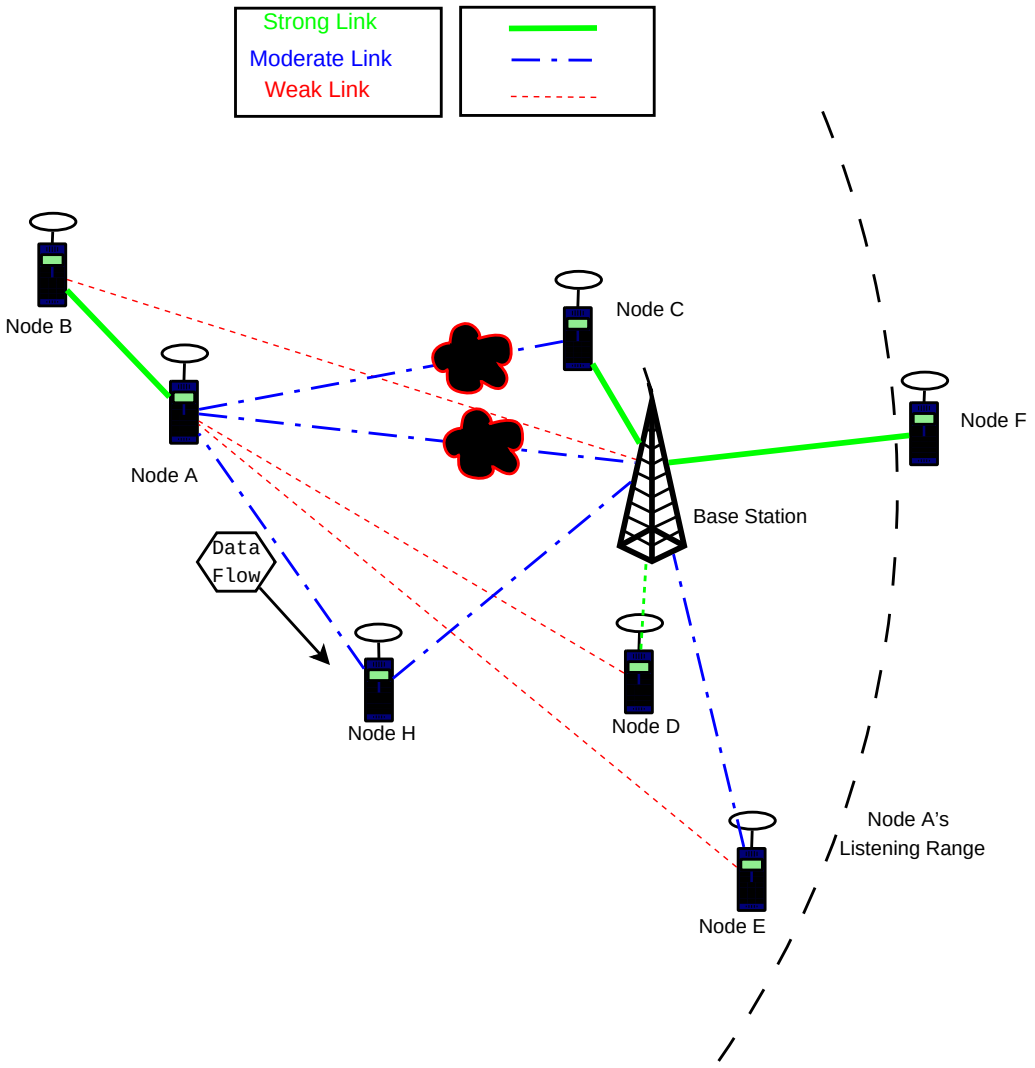

Figure 3.8: Network After Obstacle blocks A $\longrightarrow$ C link

Node A's Neighbor Tables

Optimal Relays Table

•***Node C*** ***Optimal,Unreliable***
Distance From me : 2
C's Distance from base : 2
TOTAL : 4
Link Badness : >Usable , Unreliable
C's Sink Link Badness: 0, Reliable
Power : 100 , Reliable

Best Relays table

◦***NODE H*** ***Viable Relay, Reliable***
Distance From me : 2
H's Distance from base : 2
TOTAL : 4
Link Badness : 0 , Reliable
H's Sink Link Badness: 0, Reliable
Power : 100 , Reliable

◦***NODE B*** ***Viable Relay,Reliable***
Distance From me : 1
B's Distance from base : 3
TOTAL : 4
Link Badness : 0 , Reliable
B's Sink Link Badness: 0, Reliable
Power : 100 , Reliable

◦***NODE D*** ***Viable Relay, Reliable***
Distance From me : 3
D's Distance from base : 1
TOTAL : 4
Link Badness : 0 , Reliable
D's Sink Link Badness: 0, Reliable
Power : 100 , Reliable

Neighbors Table

◦***NODE C***
Distance From me : 2
B's Distance from base : 1
TOTAL : 3
Link Badness : 0 , Reliable
C's Sink Link Badness: 0, Reliable
Power : 100, Reliable

◦***NODE D***
Distance From me : 3
B's Distance from base : 1
TOTAL : 4
Link Badness : 0 , Reliable
D's Sink Link Badness: 0, Reliable
Power : 100, Reliable

◦***NODE H***
Distance From me : 2
H's Distance from base : 2
TOTAL : 4
Link Badness : 0 , Reliable
H's Sink Link Badness: 0, Reliable
Power : 100 , Reliable

◦***NODE E***
Distance From me : 3
E's Distance from base : 2
TOTAL : 5
Link Badness : 0 , Reliable
E's Sink Link Badness: 0, Reliable
Power : 100, Reliable

◦***NODE B***
Distance From me : 1
B's Distance from base : 3
TOTAL : 4
Link Badness : 0 , Reliable
B's Sink Link Badness: 0, Reliable
Power : 100, Reliable

Figure 3.9: Routing Tables After Obstacle blocks A $\longrightarrow$ C link

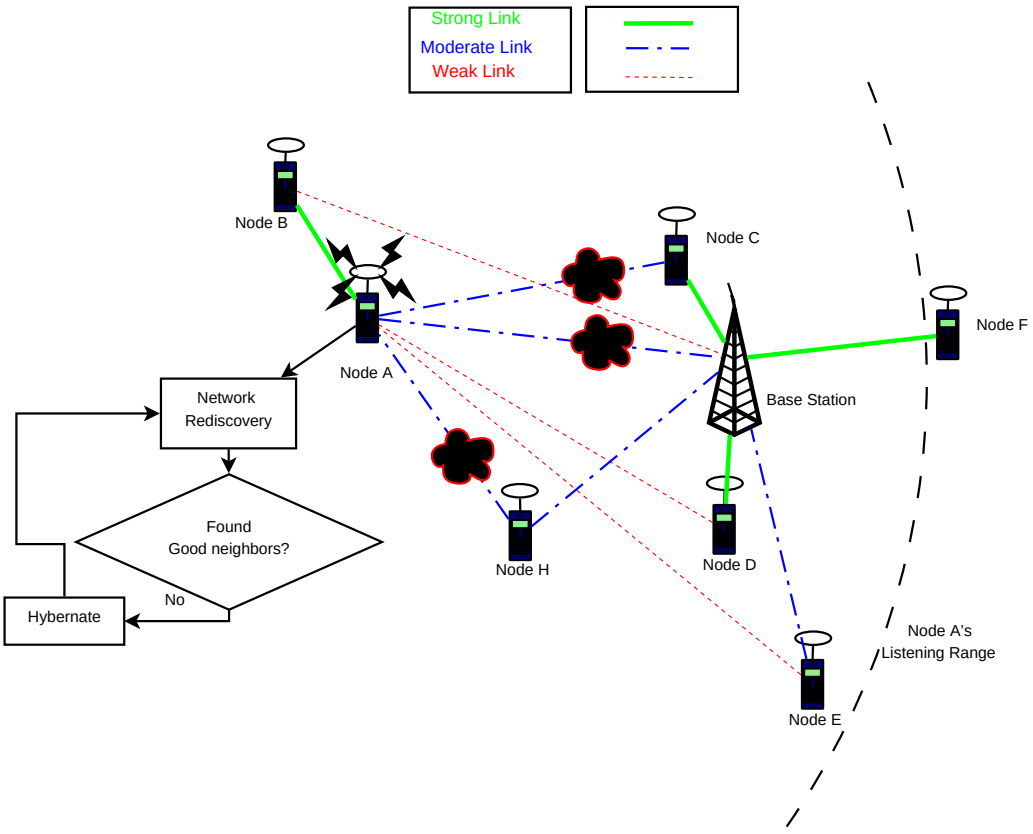

Figure 3.10: Network After Obstacle blocks A $\longrightarrow$ H link

Node A's Neighbor Tables

Optimal Relays Table

- ***Node C*** ***Optimal,Unreliable***
  Distance From me : 2
  C's Distance from base : 2
  TOTAL : 4
  Link Badness : >Usable , Unreliable
  C's Sink Link Badness: 0, Reliable
  Power : 100 , Reliable

Best Relays table

- ***NODE H*** ***Viable Relay, Unreliable***
  Distance From me : 2
  H's Distance from base : 2
  TOTAL : 4
  Link Badness :>Usable , Unreliable
  H's Sink Link Badness: 0, Reliable
  Power : 100 , Reliable
- ***NODE B*** ***Viable Relay,Reliable***
  Distance From me : 1
  B's Distance from base : 3
  TOTAL : 4
  Link Badness : 0 , Reliable
  B's Sink Link Badness: 0, Reliable
  Power : 100 , Reliable
- ***NODE D*** ***Viable Relay, Reliable***
  Distance From me : 3
  D's Distance from base : 1
  TOTAL : 4
  Link Badness : 0 , Reliable
  D's Sink Link Badness: 0, Reliable
  Power : 100 , Reliable

Neighbors Table

- ***NODE C***
  Distance From me : 2
  B's Distance from base : 1
  TOTAL : 3
  Link Badness : 0 , Reliable
  C's Sink Link Badness: 0, Reliable
  Power : 100, Reliable
- ***NODE D***
  Distance From me : 3
  B's Distance from base : 1
  TOTAL : 4
  Link Badness : 0 , Reliable
  D's Sink Link Badness: 0, Reliable
  Power : 100, Reliable
- ***NODE H***
  Distance From me : 2
  H's Distance from base : 2
  TOTAL : 4
  Link Badness : 0 , Reliable
  H's Sink Link Badness: 0, Reliable
  Power : 100 , Reliable
- ***NODE E***
  Distance From me : 3
  E's Distance from base : 2
  TOTAL : 5
  Link Badness : 0 , Reliable
  E's Sink Link Badness: 0, Reliable
  Power : 100, Reliable
- ***NODE B***
  Distance From me : 1
  B's Distance from base : 3
  TOTAL : 4
  Link Badness : 0 , Reliable
  B's Sink Link Badness: 0, Reliable
  Power : 100, Reliable

Figure 3.11: Routing Tables After Obstacle blocks A $\longrightarrow$ H link

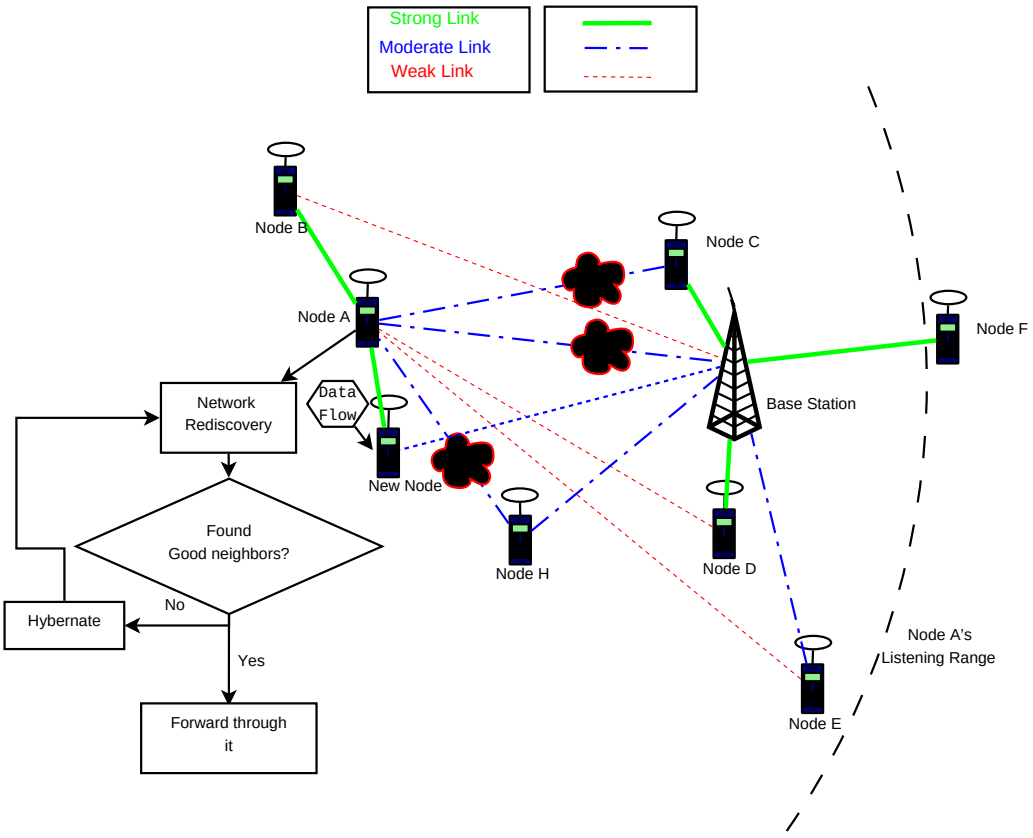

Figure 3.12: Network After New Node is found

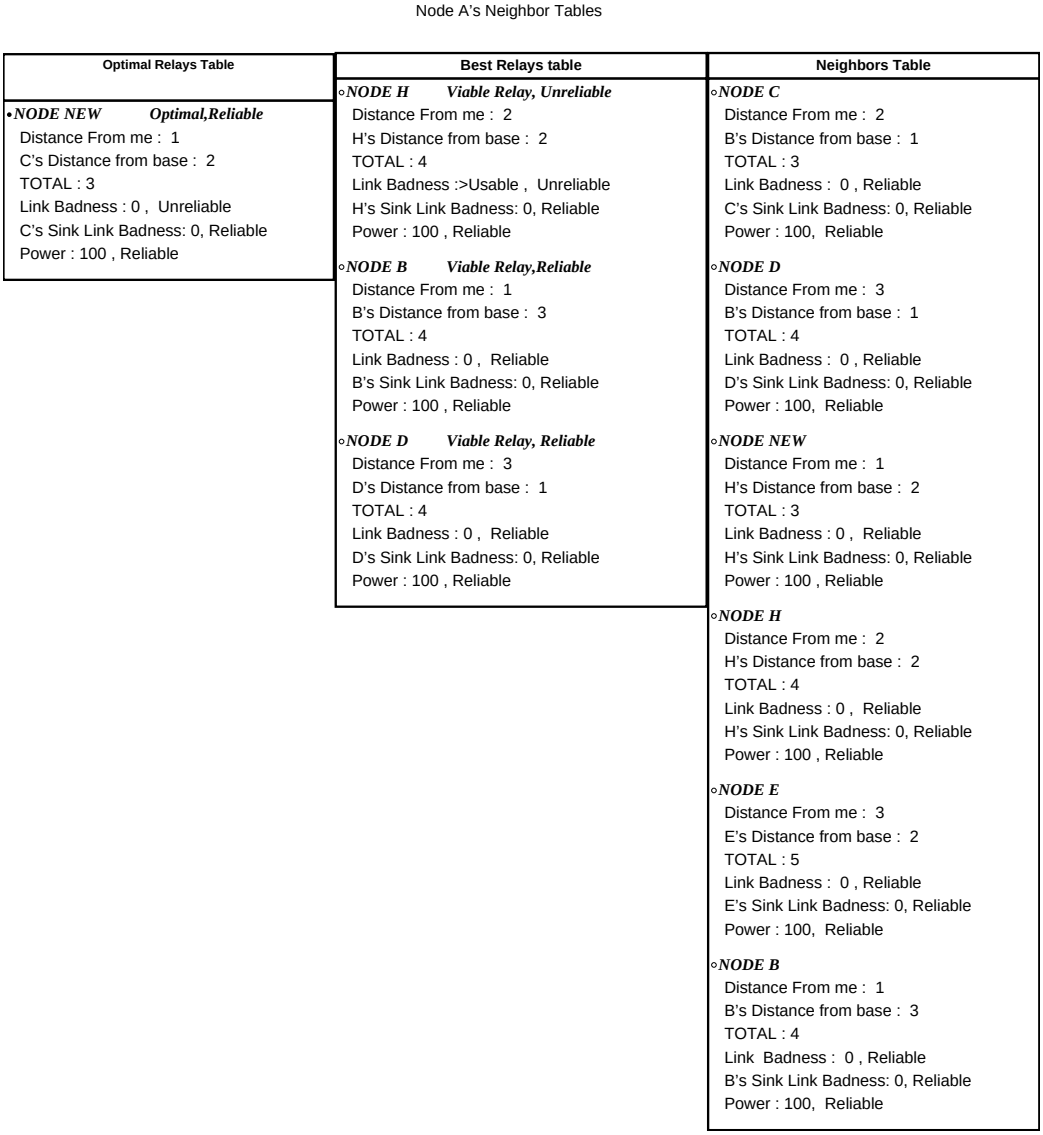

Figure 3.13: Routing Tables After New Node is found

Chapter 4

Implementation and Evaluation

In this section we will discuss the specifics of the implementation of the Designed MAC and Routing layers in the scope of this Thesis, problems encountered, solutions applied and how code optimization strategies were utilized to fit our hardware and enabled the algorithms designed to be able to run real time.

4.1 Purpose

The original purpose behind this thesis was to construct a fully fledged WSN network serving a digital garden environment. Due to time constraints though, we managed to carry out the implementation of the communication layer of the network leaving the application layer as future work.

To be able to prove the functionality of the implemented communication layer, we implemented a traffic generator that simulates the packet generation in such an environment, and some interferers to prove our networks resilience and robustness under non ideal circumstances.

4.2 Overview

4.2.1 Hardware

The complete implementation was done in the same platform described at chapter 1, with the SiLabs C8051F320[2] MCU and Texas Instruments CC2500[1] 2.4 GHz transceiver radio. The choice for this particular configuration was made fro the following reasons:

MCU C8051F320 was chosen, because it is based on the C8051 design wits befits these low power, relatively simple applications and because it had the following features:

- Low Power Consumption
- Very low Cost
- Very Well written manual and documentation
- SPI and serial Interfaces implementation

The MCU features 256 bytes of DATA ram, of which 128 are reserved for the various registers required of its run- time operations in its pipeline Architecture. The upper 128 Bytes are available to use as variable storage for any code written, and are accessed normally either with direct or indirect addressing. It also has 1024 Bytes of external flash ram able to support the various arrays and table required for the designed protocols, and 16 K Byte of code flash RAM where it accommodates the written C code that implement the functionality of the algorithms. Its clock can range from 1.5 MHz to 24 MHz, but since low power consumption is needed the lowest possible frequency was used which is 1.5 MHz.

Radio CC2500 was chosen primarily for its low cost, frequency channel versatility, and its ability to operate at the 2.4- 2.485 ISM frequency band. Another important feature was its SPI interface implementation.

Test Bed and Custom Platform The test bed used for the main development of the thesis implementation was performed at the SiLabs C8051F320 Development Kit. The platforms serving as Nodes in the final version of the implementation and the testing of this Thesis were the iCubes platforms, designed and assembled at the Technical University of Crete. They are mobile platforms featuring the aforementioned C8051F320 MCU and the CC2500 radio, along with a battery holder and a specially designed PCB board, the extension board (XT-board), that allows us to utilize the UART interface,

various pins for I/O operations and the debug adapter for programming and if need be, power.

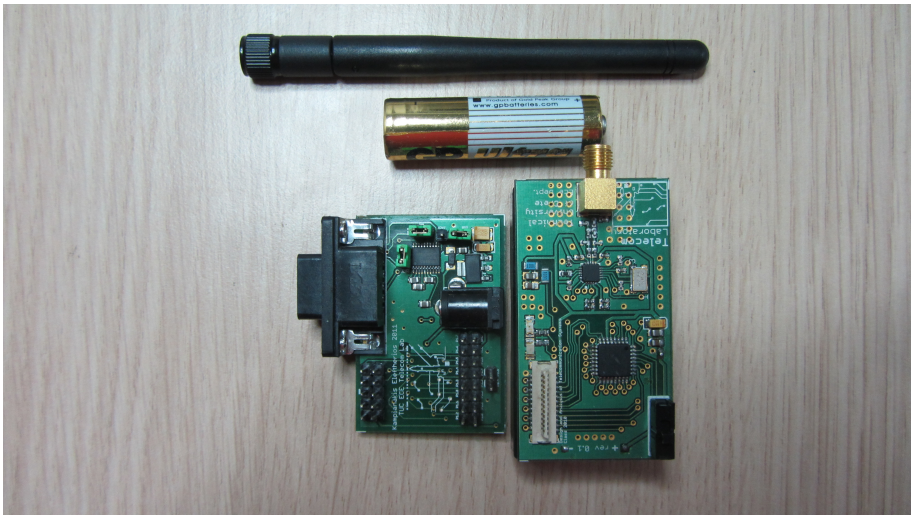

Figure 4.1: iCube with its connector extension board(XT) and antenna

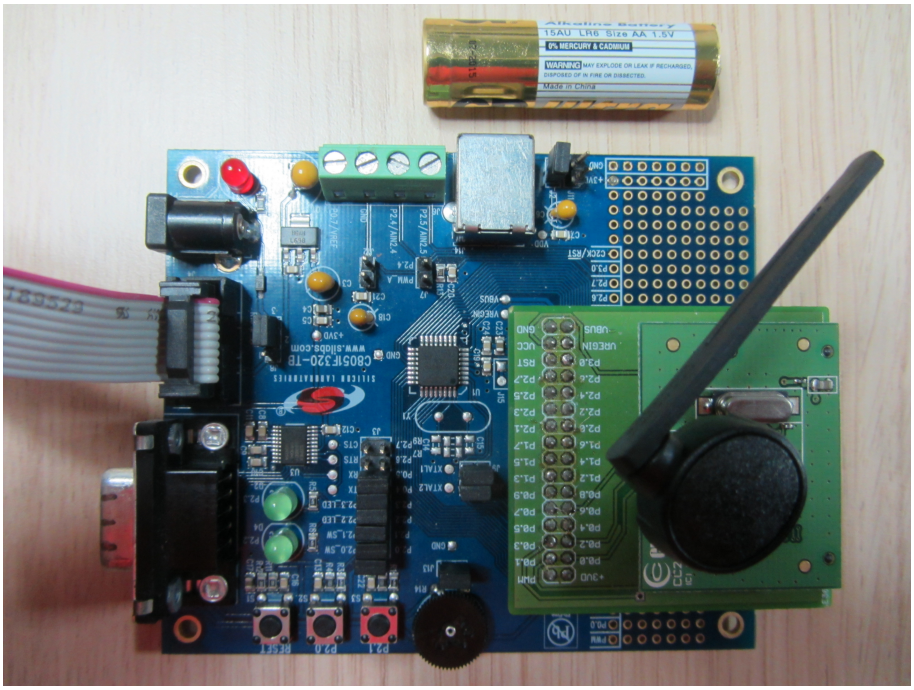

Figure 4.2: C8051F320 Development Kit connected with CC2500 evaluation module via custom interface board, dreated in TUC TeleCom Lab.

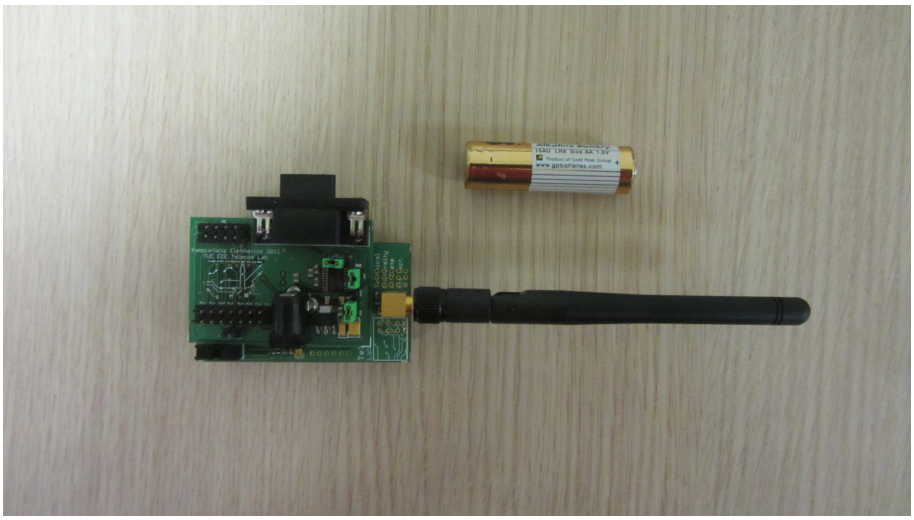

Figure 4.3: iCube with XT board, ready for use

4.3 MAC Implementation Results

Specifics: We implemented The designed MAC at the aforementioned custom made platforms using SiLabs C8051F320 MCU and Texas Instruments CC2500 low cost transceiver radio.
The tests were run in the telecommunications laboratory Of the Technical University of Crete, where up to 8 nodes where set to transmit 1000 packets each at the highest traffic possible; after each transmission successful or not, every node would repeat the procedure. Each node would transmit at about once per 50-100 ms, and each test was performed, on average, 4 times; the presented results(at figure 4.6) are the arithmetic mean of the metrics from these tests.
The goal of this tests where to produce lower bound of the performance of this Thesis' MAC, as it was designed primarily for low traffic networks. There where 2 different topology set-ups with 2 scenarios in each-fro a total or 4 different tests.

STAR topology, in which each node was within transmission range of the Base Station (Sink). as shown in image 4.4.

2 HOP topology, in which half the nodes where aligned around the base station and the other half behind the said nodes, forming a layer 2 hop network, as shown in image 4.5.

The different Scenarios:

PURE: in this scenario no fairness feature was activated, all nodes sent at maximum packet transmission rate.

FAIRNESS ON: in this scenario the following node fairness features were activated: a) *Suppression Feature:* If a node backs off too much,either due to contention loss or foreign traffic during its attempts to establish a link, it will encapsulate a command in its sent packets that forces listening nodes to sleep for an interval large enough to unsure the successful transmission of tits packet.

b)*Fairness wait:* After each successful transmission a node will wait a total of 2 packets transmission times, so as to allow other nodes to also safely transmit their data without adding to the channel's traffic.

c) *Traffic Memory:* Each node kept track of how many times it encountered high traffic during its stay on a particular channel. Each time heavy traffic was detected, anode increases a metric called Channel Traffic. The higher this indicator is, the more likely this channel will be under heavy load; so a visiting node will no longer automatically perform traditional LTB procedure, rather, it will perform probabilistic LTB as described in 1 and 2.2.

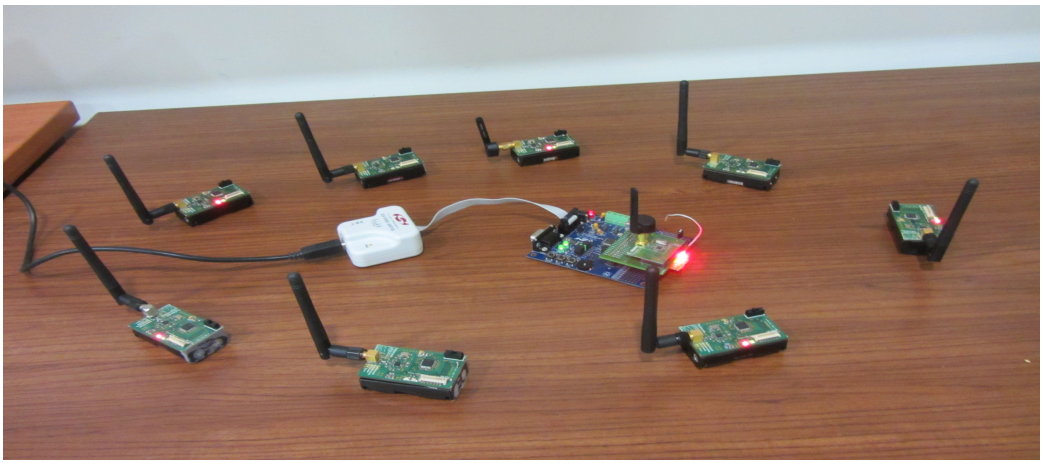

Figure 4.4: Star topology test bed

In the above and below picture the development kit in the middle acts as a base station.

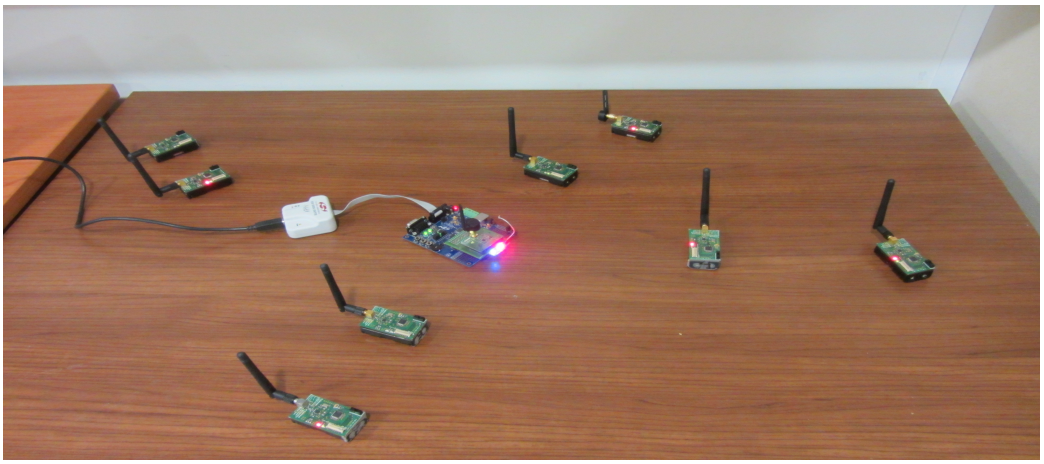

Figure 4.5: 2-HOP topology test bed

4.3.1 Results graphical representation

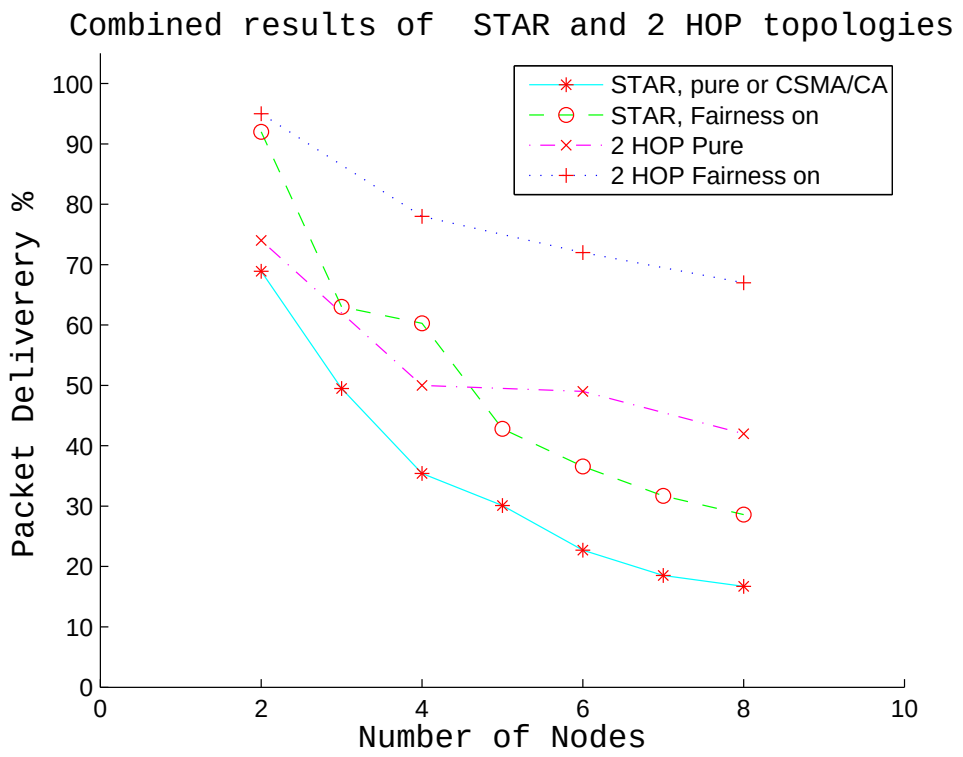

Figure 4.6: STAR - 2 HOP topologies Delivery Percentage Comparison

4.3.2 Result Analysis

The produced results pointed that the idea to utilize multiple channels to avoid channel congestion can be rewarding. Under heavy traffic and with no retransmissions per packet, as the number of continually transmitting nodes increases, the packet delivery ratio of a single frequency CSMA based protocol wanes and drops to very low points. This can be seen by figure: **??** where at the case of 8 nodes contesting simultaneously for the sink's attention, have their packet delivery ratio drop as low as 13.9. Employing a wait time of 2 packets after each successfully transmission can yield increased ratio, as high as 28 at the worst case scenario tested, bringing increased performance at the cost of added delay to the network.

When the utility of multiple frequencies was used (at the 2 hop scenario, where each of the 2nd level nodes was in a different frequency and after each transmission the 1st level nodes would jump to that same frequency to receive a new message) we can see significantly increased results, even at the case where no fairness backing off is used. With more than 4 nodes vying for the sinks attention the "unfair" multi-frequency scenario surpasses the performance of even the fair single frequency scenario by a factor of 2. The lower delivery ratio of the multi-frequency test with less than 4 nodes, can be attributed to the time-outs caused by the contention time of the "relay" nodes for the sink's attention. When the said nodes are contenting the 2nd level nodes will try to transmit new packets as they cant be sure that the relaying node has returned to their channel. This was done so we could provide with a lower limit of our MAC's efficiency.

When the same fairness strategies where employed in the multi-frequency scenario packet delivery ratio increased by a factor of 3 in the case of 8 nodes transmitting. The increased performance can be attributed to the time each node had to wait after each successful transmission, which compensated for the lack of prediction of when the relay node will be available again.

Packet delay was another significant factor, which is a traditional trade off factor in communication protocols. We observed that as the number of distance hops in the network went up, the packet delivery delay also increased,

almost proportionally to the number of hops. This can be attributed to the fact that in a multihop network a packet generated from a leaf node has to be relayed through several intermediary nodes in order to reach its destination, this added inter-node coordination significantly adds to the packet delay of the network. Another factor contributing to packet delay was the time required for a node to calculate a target frequency and hop to it as dictated by the multi-frequency nature of the network. While this extra effort from each node might seem as a significantly hindering factor to a packet's delivery speed, the increased packet delivery ratio of the multi-frequency scheme seems to outweigh this extra delay. The various tests performed with the proposed protocol required 20% extra time, but yielded approximately 3 times the delivery ratio of the traditional CSMA scheme.

4.4 Application Simulation

After testing the designed MAC under the non-ideal circumstances described, we went on to crate a real application by simulating sensory readings by uniform random number generator. Every second there was a 15 per cent chance of a new packet arrival which required immediate transmission; a relatively high traffic scenario for a WSN.

Network description and topology: Our net work consisted of 8 nodes all trying to forward their generated data to the sink using the designed MAC and routing protocols. The Nodes had not prior knowledge of the topology; they had to discover the network and organize themselves into clusters of neighbourhoods as described in: 3.3 at page; 45

Simulation Description The simulation was left to operate uninterrupted for one day inside the Telecommunications laboratory of TUC. The Nodes were arranged in a 3-hop network with no prior knowledge of the topology. The nodes arranged themselves in a network automatically and dynamically forwarded their packets using the protocol proposed in this thesis.

As the network's nodes gradually depleted their power, relays that for-

warded the majority of the data, saw their data load gradually decrease at the rest of the nodes sought other sub-optimal choices to forward their data; a direct result of the power sensitive feature of the routing protocol. The experiment was concluded after each node had successfully forwarded 1000 self generated packets, forwarded packets did not count towards the 1000 mark.

From the tests run, we observed that out of the 1000 packets generated and sent less than 50 where lost, counting the packets lost during the initial setup and the various periodic routing exchanges, indicating a 95% total packet delivery chance. It is interesting to note that pure data-packet loss was less than 3%.

4.5 Difficulties Encountered

4.5.1 Theory

Design: The most difficult problem encountered during the span of this Thesis was to design a mechanism for updating traffic,interference or,relay badness that guarantees a level of cognitiveness while being simple enough to be updated constantly and fast enough so its does not dilapidate real time communication. We had to come up with a way with which when a bad channel or a neighbour would be available again without checking or polling, in order to conserve energy. with all these constraints forming a demanding burden we turned to the concept of temporal correlation: a channel or a neighbour found increasingly bad, is not likely to be available any time soon. We made that connection thinking, that if an obstacle or a malicious interferer partitioned the network it would not disappear shortly after. So we designed 4 different level for these metrics, each increasingly making the assumption that the particular network resource will not be available any time soon. So at the lowest level (1), a resource's badness increases and decreases at the same rate, at the next two levels the badness increase out paces the decrease. at the final level the resource is deemed inaccessible, thus any breakthrough of communication toward it greatly diminishes its badness

(namely to 50%).

Synchronization: A major problem in the whole concept of the MAC and routing protocols was node synchronization. More specifically, how could we guarantee that at the average case scenario a node would be able to locate within reasonable time. Being asynchronous in nature, it is quite frequent for a node (say A) to try and communicate with another node (B) whilst the later is in the middle of another data exchange. In such a case the recipient node (B) would not be in a position to receive A node's packet, and node A would also not be able to determine the cause for the lack of answer from Node B; it could be due to partitioning, B's death, another data exchange e.t.c.

Solution: To counter this ambiguity and the problems it brought along, a metric was created, *Node Badness* that show how reliable is a node. The higher the badness the more likely that a packet will not be successfully received.

4.5.2 Implementation

Code Size A significant constraint during the implementation phase, MCU's C8051F320 16 K byte code size memory meant that each code segment that needed to be written, had to be optimized and as compressed as possible. This lead to much time spent think on how to utilize the KEIL compiler to save memory space. Each part of every algorithm had to be rethought, even if the functionality was already designed, so as to be written with as few lines as possible. This lead to a relatively condensed code with a great deal of commendation.

Code density The dense code meant that each line had increased significance, leading to increased periods of thinking during each debug attempt. In order to avoid long thinking hours, we crated templates for specific functions such as loop controls, regular expressions checking, function calls and declarations and other control mechanisms in general. Also each function

was separated into logical parts in order to have clear, structured code which made debugging and improvement integration possible.

Code Architecture: The code has to be written in a modular, clear, structured way not only because of the increased complexity of its logic and size, but also because during the implementation and testing of the algorithms, many new problems and improvement opportunities arose. All these new hindrances and boons had to be able to be easily integrated with the rest of the code without major overhauling, which would lead to a lot of time wasted in trivial problems. The architecture of the code was one of the most important issues of this thesis, as in the early stages that both skill and experience on embedded programming were lacking, a lot of time was spent on trivial matters such restructuring or even rewriting badly

MCU specific: While programming the C8051F320 MCU many problems arose, using up a fair amount of time for their solution. First and foremost was the external data usage and initialization. F320 has 1 K byte external data, that one can access by simply declaring xdata before a variable name. A mishap about this xdata is that at times, if it is defined and declared at the same line, and that is done globally, the MCU control crashes offering no warning. One can download a seemingly correct code, but if a global xdata stored variable is defined and declared at the same line, the program does not work.
Solution: the answer to this problem is to declare and define an xdata variable at separate points.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

By designing, implementing and testing a new asynchronous, cross layer multi-frequency communication protocol, we were able to gain a first hand experience on how utilizing multiple frequencies for communications schemes can impact on overall performance. Based on our results from the scenarios tests in 1 and 2 hop topologies, we were able to see that being able to communicate using multiple orthogonal frequencies can lead to increased performance in the sense of packet delivery ratio and interference resistance. The network working with the communication protocol proposed in this thesis, was able to remain functional despite interference or physical partitioning problems, due to the cognitive nature of the protocol's routing layer. We were led to increased packet delivery ratio's under heavy traffic situations, where the same circumstances were debilitating to an asynchronous, single frequency scheme such as the common CSMA/CD. Having our own implementation on custom built nodes, the iCubes, granted us valuable insight on some of the most common and persistent problems one can encounter when trying to transfer a conceived protocol into a realistic environment, which can be used as a basis to further improve the proposed protocol.

In closure, we were able to develop an adaptive and cognitive communication protocol that handles with acceptable performance the cornerstones of the MAC and routing layers, while being able to work around the major problems of interference and partitioning, that can be used as the foundation upon which a fully fledged Wireless Sensor Network could be built.

5.2 Future Work

The variety of layers and problems encountered and addressed during the design and implementation led to a great many deal of ideas for future work to extend or improve the mechanisms proposed in this thesis. We categorized these ideas based on the part on which the apply into three groups: a) Hardware, b) Software and c) Algorithmic.

We will now proceed list and expand somewhat on each notion to provide with a perspective on what could still be done.

Hardware Future Work Implementing the designed MAC and Routing protocols into the C8051F320 MCU was a challenging task, due to the low data and code memory of the particular processor. It quickly became evident that in order to use the protocols into a real application that needs to perform more than just collecting sensory data, controlling a few simple devices and employing our communication proposal, we need an MCU with more specious memory as the code size implementing the MAC and Routing needs 14.5 K bytes our of 16 k Byte and utilizes 100 bytes out of 128 availing for the MCU, despite the sophisticated over layer's efforts. While the 100 bytes of data ram is not a problem, because with careful programming and explicit commands to the over layer process one can implement a lot of functionality spread to many different functions and not increase the data ram usage by a single byte; the 1.5 K byte left for code usage limits things greatly. Another added benefit or porting the MAC Routing code to a more powerful low cost, low power MCU is the modern processors capability of lowering their clock frequency to just a few kHz which would further reduce energy consumption making a network made for such nodes even more robust.
An interesting future work proposal would be to port said codes to C8051F1003 newly acquired MCU by the TUC Telecom laboratory.

Software Future work A very interesting and important part of the implementation work and one of the fields that could yield impressive results if optimized.

Due to the size of the implemented code some compromises had to be made to favour simplicity over code efficiency. While great care was taken to write according to the compilers direction's and embedded programming rules, some points where written in such a way as to provide easy reading and understanding due to the constant debugging, which was also the main time sink during the implementation phase. Also some of the features provided by the CC2500 radio's hardware where implemented in software instead, as the point of this phase was to provide a proof of concept and activating the radio's functioning had some increased complexity requiring significant time and testing.
By further optimizing the written code and activating the radio's built in hardware to replace the software currently handling some features such as receiving time-outs, would bring added processing speed and reduced energy consumption.

Algorithmic Future Work The last and most important category containing all of the ideas discovered in the process of this work to optimize both the MAC and Routing layers. Here after follows a list and a short description on each one of these new aspects:

Badness Indicators fine tuning. Whether updating the traffic indicator of a channel, the badness of node or a channel, studying and optimizing how these metrics should increase or decrease would improve the performance of the protocol.

Channel Probing Some as above, fine tuning the mechanisms of when to check if a previously bad channel is found free, or scout for a base station would lead to increased performance.

Real time clock implementation. While the concept of the MAC layer protocol was to be an asynchronous one, if one moves to more expensive MCU with accurate crystal it would interesting to see if creating and integrating a real time clock to the protocol would bring added benefits. The main idea is to be able to predict when a node would wake up, so

as to minimize the energy spend on idle receiving as much as possible. By being able to predict when a node would wake up, every node in the network would have their radios turned off to conserve power; even more than just low power listening.

Cluster forming. At the current point nodes form a virtual cluster based on frequency, but also hold information about their neighbours relative distance, thanks to the routing algorithms. By using these cross layer features together a node could form real cluster, with a leading node and its children, based on relative distance and power remaining.

Broadcast fine tuning. Although the broadcast mechanism is implemented, further optimization if possible by creating a mechanism that each node that receives a broadcast message from a sink could pass it down its children if clustering, as explained above is implemented.

Appendix 1

Bibliography

[1] Texas Instruments *CC2500: Low Cost, Low-Power 2.4 GHz RF Transceiver*

[2] SiLabs *C8051F320 Low Cost, Low-Power MCU*

[3] Lei Tang, Yanjun Sun, Omer Gurewitz, David Johnson. *"A Dynamic Multichannel Energy-Efficient MAC Protocol for Wireless Sensor Networks"* MobiHoc, May 16-19, 2011, Paris, France.

[4] L. van Hoesel, P. Havinga, *"A lightweight medium access protocol (LMAC) for wireless sensor networks"*, In Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS 2004), Tokyo, Japan, June 2004.

[5] Zhou, Gang, Chengdu Huang, Ting Yan, Tian He, John A. Stankovic, and Tarek F. Abdelzaher. *"MMSN: Multi-frequency media access control for wireless sensor networks."*, In IEEE Infocom, pp. 1-13. 2006.

[6] Kim, Youngmin, Hyojeong Shin, and Hojung Cha. *"Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks."*, In Proceedings of the 7th international conference on Information processing in sensor networks, pp. 53-63. IEEE Computer Society, 2008.

[7] Wei Ye, John Heidemann, Deborah Estrin *"An Energy-Efficient MAC Protocol for Wireless Sensor Networks"*, Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, IEEE, New York, USA, June 23-27, 2002.

[8] Manish Kumar Jha, Atul Kumar Pandey, Dipankar Pal, Anand Mohan *"An energy-efficient multi-layer MAC (ML-MAC) protocol for wireless sensor networks"*, AEU-International Journal of Electronics and Communications 65, no. 3 (2011): 209-216.

[9] Tobagi, Fouad A. *"Random access techniques for data transmission over packet switched radio networks"* No. UCLA-ENG-7499, California University, Los Angeles, School Of Engineering And Applied Science, 1974.

[10] Polastre, Joseph, Jason Hill, and David Culler. *"Versatile low power media access for wireless sensor networks."*, In Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 95-107. ACM, 2004.