Transfer Thesis

Noradila Nordin

Department of Electronic & Electrical Engineering

University College London

Torrington Place

London

WC1E 7JE

noradila.nordin.12@ucl.ac.uk

February 8, 2016

Abstract

This report presents the current state of research work that has been carried out in the context of the PhD. (describe why do it). The PhD work proposes a new decentralised multi-channel tree building protocol with a centralised controller for ad-hoc sensor networks. The protocol alleviates the effect of interference which results in improved network efficiency and stability, and link reliability. The proposed protocol takes into account all available channels to utilise the spectrum and aims to use the spectrum efficiently by transmitting on several channels. The protocol detects which channels suffer interference and changes away from those channels. The algorithm for channel selection is a two-hop colouring protocol that reduces the chances of nearby nodes to transmit on the same channel. All nodes are battery operated except for the low power border router (LPBR). This enables a centralised channel switching process at the LPBR. The protocol is built based on the routing protocol for low power and lossy networks (RPL). In its initial phase, the protocol uses RPL's standard topology formation to create an initial working topology and then seeks to improve this topology by switching channels. The report discusses the main engineering and research challenges raised by the protocol, and describes and explains the principles and mechanisms used to support the proposed protocol. It then presents an extensive evaluation of the protocol and other other approaches. The implementation and evaluation of the protocol is performed using the Contiki framework. The report then describes the future main research issues that will be investigated in the context of this PhD.

In this report,

The proposed approach The report discusses

Acknowledgements

Acknowledge all the things!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivation

Wireless Sensor Networks (WSN) are ad-hoc networks that consist of sensor nodes that typically use low power radios such as IEEE 802.15.4, a relatively short range transmission standard radio technology in the 2.4 GHz band. The standard allows transmission to occur on several different channels within this band [1]. Unfortunately, the channels used by this technology often suffer interference [2, 3], for example, from Wi-Fi [4, 5] and Bluetooth. Sensor networks have to contend with an increasing number of devices that cause this wireless interference. Organising the network topology around this interference becomes an enabler for increasing transmission efficiency at a smaller energy cost. WSNs need to be able to operate reliably in the presence of such interference. It is important to minimise energy costs in these networks since deployments can be for weeks, months or longer.

Multichannel communication in wireless networks can alleviate the effects of interference which, as a result, can improve the network efficiency and stability, link reliability and minimise latency [6]. It also enables communication between physically proximate nodes to occur simultaneously without the risk of collision when the communicating nodes use different channels. However, not all channels are free from interference. Some channels would perform better than the other channels depending on the current location and network environment. Therefore, nodes should consider hopping to another channel when the channel shows an alarming decline

in performance.

1.2 Problem Statement

It is clear that it is impossible to find a single channel guaranteed free from interference and there is no consensus on the best channel to use. The work in this PhD introduces a multichannel protocol that takes into account all available channels to utilise the spectrum and checks the condition of the channels before hopping to avoid those channels with interference. Several previous studies have developed multichannel MAC layer but, despite the potential benefits none are yet widely implemented in real world deployments.

1.3 Contribution

Important aspects of this work is in improving the WSNs performance by implementing multichannel routing protocol to investigate the effect in comparison to a single channel in a noisy and lossy environment. There are several research challenges in a multichannel protocol design in term of the synchronisation, topology formation and maintenance, channels initiation, channel selection processes and the energy consumption of the multichannel protocol. The multichannel protocol in this work consists of centralised and decentralised parts where the centralised node controls the network and the decentralised node carries out the multi channel processes. The work in this PhD investigates the channel selection processes based on the nodes cross layers interactions during decision making.

1.4 Current Work

Multichannel Cross-Layer Routing Protocol (MCRP) has been developed as a new multichannel protocol. MCRP is proposed to enable communications on all available channels in the spectrum to avoid interference, congestion and conflict in the network. MCRP consists of two main parts; a centralised intelligence at LPBR, and decentralised nodes. LPBR implements a two-hop colouring algorithm to avoid interference between physically proximate nodes trying to communicate on the same channel. The information on channel interference and network topology from the

lower layer is made available to the application layer. This allows the centralised controller (LPBR) to have an overall view of the system to make decisions at the network and MAC layers about which channels nodes should listen on. The system is fail safe in the sense that the WSN functions if the central system which assigns channels fails temporarily or permanently. MCRP avoids channels with interference which greatly reduces the effects of interference on the network. MCRP is implemented in Contiki [9], an open source operating system for WSNs. The performance of MCRP has been evaluated in Contiki network simulator, Cooja [10] and Flocklab [81] testbed.

1.5 Report Outline

The remainder of the report is organised as follows. Chapter 2 introduces the state-of-the art in the area of multichannel protocols. It also presents the main current research efforts towards () Section **??** presents related work to multichannel protocols. Chapter 3 presents the main features and mechanisms used in MCRP. It describes (). It also presents (). Section **??** describes the key idea of our proposed protocol and the high-level design, and the implementation of the protocol in Contiki. We describe and evaluate the experimental results in Section **??**. Chapter 7 summarises the current work and presents the future research works that will be investigated in the context of this PhD.

Chapter 2

Literature Review

2.1 Wireless Sensor Networks

A WSN is a network of sensor nodes that are used to collect data from the target area over the radio. These data that the sensors send can be normal data packets or sensor measurements data such as the temperature and movement in the specific area where the sensors are located. The sensors can be used for continuous sensing, event detection, location sensing and local control of actuators to control different components in the sensing device, such as adjusting the sensor parameters or move the sensor if it is a mobile sensor.

This chapter describes the available WSN applications, challenges and known issues that occur in WSNs. Many previous studies were done in order to maximise the lifetime of sensor networks while keeping the energy to a minimum. This chapter also briefly describes the existing solutions for energy efficient multi channel at the MAC and network layers which prompted to the work of MCRP.

2.1.1 Applications Overview

There are five types of deployed WSNs: terrestrial WSNs, underground WSNs, underwater WSNs, multimedia WSNs and mobile WSNs which cover different types of environment; to deploy on land, underground and water [11]. Unlike other sensor nodes, multimedia WSNs have the ability to monitor and track events in the form of video and audio as they are equipped with cameras and microphones for multimedia data which can enhance the existing WSN applications [12]. Mobile WSNs

on the other hand, can be any type of sensors that have the capability to reposition and organise itself in the network.

WSNs evolution is driven by a number of emerging applications that focuses on the importance of wireless sensors in applications such as smart grid, areas in smart cities, and automated home, building and industrial applications [13]. Smart grid could save considerable amounts of energy by improving the existing electrical grid power. Smart cities which includes automated home, building and industrial in populated cities can improve the environment quality by allowing automated services such as pollution monitoring and automated energy control (temperature and lighting) which increases the energy saving in the process.

WSNs applications are important as sensor nodes can be easily deployed at all types of environment, installed and require minimal maintenance for a period of time. The main challenges in these applications are in term of reliable event detection, securing high data rates for efficient data routing and dense or sparse nodes deployment.

WSNs applications can be categorised into five main monitoring and tracking applications which are the environmental applications, health applications, home applications, military applications and other commercial applications [14]. These applications are briefly described in the next section with examples for each category.

2.1.1.1 Environmental Applications

The environment applications can be divided into two types; tracking and monitoring. The tracking applications are used to record the movements of animals such as birds, insects and small animals at a certain area. Monitoring applications are used to monitor the environment conditions such as forest fire detection, flood detection, biocomplexity mapping [15], precision agriculture monitoring and volcanic monitoring [16].

In forest fire detection, the sensor nodes are used to relay the exact originated location of the fire to the end users to control it from spreading. ALERT is an example of a flood detection system that is deployed in the United States. ALERT

consists of several types of sensors such as rainfall, water level and weather sensors. In agriculture, the sensor nodes are used to monitor the level of pesticides in drinking water, soil erosion and air pollution in real time. In volcanic monitoring, the sensor nodes allow measurements to be taken from locations that are otherwise inaccessible.

2.1.1.2 Health Applications

Sensor networks in health applications can be used to monitor human physiological data such as detecting elderly people's behaviour in case of a fall, drug administration [17] in hospitals to minimise incorrect prescription of medication to patients, and to monitor and track doctors and patients locations in a hospital. Examples of these are *telecare* and *telehealth* [18].

Telecare is a system of wireless sensors that are placed around the house and can be a personal alarm in form of a small wristband or pendant. These sensors can detect risks such as a fall, motion sensor that turns on the lights at night when someone get out of bed, a pressure mat on the mattress to sense if someone gets back to bed or a sensor on the door in case it is not closed, are a few examples of the system. If a risk is detected, it sends the alert immediately for attention to a telecare monitoring centre.

Telehealth is a small equipment to monitor health from home. It can be used to measure the blood pressure, blood glucose levels, oxygen levels, weight or temperature. The measurements are automatically transmitted to a monitoring centre. The healthcare professional will be contacted if the information raised an alarm for actions to be taken.

2.1.1.3 Home Applications

In home automation [19], the smart sensor nodes and actuators can be buried in the appliances such as vacuum cleaners, microwave ovens and refrigerators which allows them to form an interaction through the Internet. Some of the recent home automation are *Samsung SmartThings* and *Nest Thermostat*.

Samsung SmartThings allows devices at home to be monitored and controlled from a mobile phone such as controlling the thermostats and lighting. Nest Ther-

mostat is a self-learning thermostat that is consists of activity sensors, temperature sensors, humidity sensor and a Wi-Fi radio. These sensors allow Nest to learn the heating and cooling habits which allows it to shuts down due to inactivity to conserve the energy. Nest is weather aware. It uses its Wi-Fi connection to get the weather condition and forecasts, and integrate the information to understand the affects of the outside temperature to the energy usage. Nest is also able to connect with other appliances that are Nest supported. The appliances can automatically start without any need to program it as it learns from other devices.

2.1.1.4 Military Applications

WSNs are used in military applications to monitor friendly forces, equipments and ammunitions by attaching sensors which report the status back to the base station; battlefield surveillance by covering critical terrains, routes, paths and straits with sensors and reconnaissance the opposing forces; assess battle damage, and to detect nuclear, biological and chemical attack by deploying sensors to explore areas and serve as warning systems to avoid casualties.

An example of military application is *PinPtr* [20]. PinPtr is an experimental counter-sniper system. It was developed to detect and locate shooters by measuring shot time of arrival of the muzzle blasts and shock waves from the sensors that are densely deployed. The measurements are routed to the base station where the shooter's location is computed. PinPtr was demonstrated and evaluated in realistic urban environment from various US Army test facilities.

2.1.1.5 Other Commercial Applications

Other available commercial applications are environmental control in office buildings such as controlling the air flow and temperature for different part of the building; car thefts monitoring and detection within specific region; inventory control management to track and locate the inventories in the warehouses; machine diagnosis in order to predict equipment failure for maintenance through vibration signatures gathered by sensors [21]; and vehicle tracking and detection for parking purposes such as the *Smart Parking* from *Streetline* and *SmartPark*.

Smart Parking solutions are used in more than 40 cities and universities in

North America and Europe. The system could make intelligent decisions using the data from the real time and historical analytical reports to improve the parking ecosystem. The system detects vehicle occupancy in real time which simplifies the parking experience by guiding drivers to the available spaces. It can also guide officers to unpaid violations and overstays as the arrival and departure times are recorded; and to detect if a car is parked over the no parking and restricted zones.

SmartPark is a parking solution in the UK, currently operating in Birmingham and in the central London Borough of Westminster. These applications enable drivers to find vacant space within the busy town and city centres quicker.

2.1.2 WSN Challenges and Issues

WSNs are widely used in various kinds of applications. This is because sensor nodes can be densely deployed, easy to install and require minimal maintenance over a period of time. However, WSNs suffer from limited hardware resources which only allow limited computational functionalities to be performed. It also suffers from limited energy capacities as the sensors are battery powered and they will become faulty and not able to function once the certain threshold of energy level is reached. It also operates in an unreliable radio environment that is noisy and error prone which drain the sensors batteries at a higher rate.

These constraints have a major impact on the sensors performance. In order to prolong the sensors lifetime thus, the network lifetime, the sensors need to be able to cope with the limitations and be as energy-efficient as possible to guarantee good overall performance.

2.2 Maximising Lifetime and Minimising Energy

In WSNs, it is necessary to estimate the nodes power consumption before they are deployed to enable accurate forecast of the energy consumption. The estimations are used to determine the nodes lifetime before maintenance and batteries replacements are required in order to have a functional network. Unfortunately, the node lifetime is very dependent on the radio environment that can be unstable, noisy and error prone which makes energy consumption to vary [22]. The network lifetime,

however, depends on various factors such as the network architecture and protocols, channel characteristics, energy consumption model and the network lifetime definition. In order to increase the network lifetime, these information regarding the channel and residual energy of the sensors should be exploited.

There are various definitions of network lifetime that have been used. These definitions are application-specific as some applications might tolerate a considerable number of loss nodes, while some applications require a higher number of nodes which any loss is considered critical to the network such as in sparsely deployed nodes of an area. The definitions impact the performance differently, depending on the applications. The various definitions are:

- **The first node to die** - The network lifetime is defined as the first node to fail in the network [23, 24]. In [25], the simulation ends when a node reaches the energy level of zero.
- **The number of alive nodes** - The network lifetime is the number of remaining nodes as a function of time. The network has a longer lifetime with a higher number of remaining nodes [26, 27].
- **The number of nodes still connected to the sink** - The network is alive based on the remaining number of nodes to have coverage to connect to the sink [27].
- **The fraction of alive nodes** - The network lifetime is defined by the percentage of surviving nodes above a threshold.
- **Packet delivery ratio** - The network lifetime ends when the packet delivery ratio drops dramatically. GAF [28] uses this definition where it is possible when the traffic is kept constant.
- **The first failure in data transmission** - The sensor does not have enough energy for transmission [23].

Network lifetime is strongly related to the remaining energy of all nodes. However, maximising the minimal energy of all the nodes is not the best way to prolong

the network lifetime as it will place heavy burden to the key nodes such that nodes that are close to the sink. These nodes drain their batteries quicker than other nodes which as a result, shorten the network lifetime.

There are four ways that have been explored from many studies to maximise the network lifetime, which are by introducing (i) energy efficient MAC protocols, (ii) energy efficient routing protocols, (iii) controlling the transmission power and (iv) using energy harvesting. These options are described in details in the next few sections, introducing the differences and advantages, and the existing proposed solutions.

The aim of the WSN design is to extend the network lifetime under the given energy and node constraints without jeopardizing reliability and communications efficiency of the network.

2.2.1 MAC Protocols

Many energy efficient MAC protocols have been proposed to prolong the network lifetime. The radio module that is controlled by the MAC protocol is the major energy consumer in WSNs. The radio uses nearly the same energy in all active operation modes such as the transmit, receive and idle modes [22]. Thus, it is important to reduce the radio usage to conserve the nodes energy.

The main causes of energy consumption are nodes collision, overhearing and idle listening [29, 30]. Collision happens when nodes that is within each other transmission range transmits simultaneously. The energy used in the collided transmissions is wasted as none of the nodes would receive the transmitted packet. Multi channel is one of the solutions to overcome collision. Overhearing happens when a node receives irrelevant packets or signals that are not intended to the node. As the radio uses nearly the same energy for all operations, this drains the node energy unnecessarily. In idle listening, the node keeps its radio on while listening to the channel for potential packets. The node does not know when it will be the receiver of the packet. Considerable amounts of energy are wasted as the node keeps its radio on for a longer period listening to an idle channel when it does not receive or transmit packets.

A vast number of energy efficient MAC protocols have been developed to overcome these problems through *duty cycling*. Duty cycled MAC protocols allow the node to periodically alters the sleep state and listen state. By lowering the duty cycle, the node sleeps for a longer period instead of being permanently active. However, the node needs to have frequent check interval to avoid deafness problem while keeping overhearing to a minimum. This reduces the energy consumed by idle listening and overhearing.

Many MAC protocols such as YMAC [31] uses duty cycle as the indicator to evaluate the energy efficiency performance. This is because it is difficult to measure the nodes energy consumption. However, there are studies that managed to estimate the energy consumption. This is described in Chapter 6.

2.2.2 Routing Protocols

Various energy efficient routing protocols for WSNs have been proposed and developed to ensure efficient packet delivery to the destination. The strategies that are used in routing protocols should ensure minimum energy consumption in order to prolong the lifetime of the network.

A major issue in WSNs routing protocol is in finding and maintaining the optimal routes that are energy efficient. This is due to the energy constraints and unexpected changes in node status such as node failure or unreachable. This causes the topology to be altered frequently to adapt to the changes. Abrupt topology modification is important to avoid the network from being disconnected which leads to higher rate of packet loss at the involved nodes as the routes are not updated.

There a several routing techniques such as flat, hierarchical and location-based routing protocols that are application dependent. Hierarchical structure, as an example, has a balanced energy structure as the packets are transmitted from the lower layer nodes to the upper layer nodes. These different techniques are explained in detail in Section 2.4.

The routes that are formed are based on the routing metric [32] that attempts to transmit the packet to the receiver by selecting the most efficient path that the protocol calculated. The path may be the shortest path, lowest expected transmis-

sion count path [33] or path that maximises the network lifetime by considering all nodes remaining energy. However, in order to achieve the best network lifetime, the total energy consumption of the network and the nodes minimal remaining energy should be combined for a better balance in the network [24].

2.2.3 Transmission Power Control

Topology control term has been used to mean two different things in WSNs literature. Several authors define topology control as routing protocol techniques. Another definition of topology control is power control techniques which act on the nodes transmission power level [34]. Topology control term has been interchangeably used with power control. To avoid confusion, the term power control is used in this thesis.

In power control, a node has control over the transmission range of the node's radio which can be manipulated to benefit the network. The power adjustment approach allows the node to vary the transmission power thus range to form a connected network that minimise the energy incurred in transmission. The nodes collaboratively adjust to find the appropriate transmission power which enables the nodes to transmit at a lower transmission power than at the maximum. However, a sparse network would require a higher transmission power than a dense network to be able to transmit to the nearest node.

The power control technique eliminates links that are wasting the energy resources by fixing the area of coverage thus routing. This reduces collisions as inefficient links of long distance nodes are discarded. However, the nodes need to change the transmission power to adapt to any area coverage changes in order to modify the routing.

As the transmission ranges are relatively short, the nodes can simultaneously transmit packet without interfering each other, thus reducing congestion from retransmissions. Although power control improves the network traffic flows, it does not reduce the nodes power consumption as it depends on the radio duty cycle. Power savings due to transmission powers are negligible [29].

2.2.4 Energy Harvesting

As mentioned previously, sensor nodes have limited energy capacities as they are battery powered. However, the number of deployed nodes within the specific area has an effect to the nodes energy usage. In a densely deployed nodes area, short range transmission between the nodes could reduce the energy consumption while a sparsely deployed nodes area have a longer range transmission which require higher energy usage. In the situation where the nodes are not densely deployed, energy harvesting may be an option to increase the nodes energy level.

Energy harvesting is when a node tries to replenish its energy by using other energy sources such as solar cells [35, 36], vibration [37], fuel cells, acoustic noise and a mobile supplier [11]. Solar cell is the current mature technique to harvest energy from light. There is also work in using robots as mobile energy supplier to deliver energy to nodes. This allows a longer network lifetime as the node has restored its energy.

However, energy harvesting depends on various environment factors such as light, vibration and heat to be generated and converted to the usable electrical energy. There are also other different powering mechanisms that are available such as rechargeable battery with regular recharging from the sunlight [29].

2.3 Multichannel MAC Protocol

In single channel MAC protocols, nodes are configured to use a single channel throughout the nodes lifetime. Frequency agile MAC protocols on the other hand, allow the nodes to switch to different channels during run time. This is possible as recent radio chips take less than 100μs to switch to a different channel. The channel switching delay is negligible which attracts multi channels to be used in WSNs. Multi channels have the advantage of an increase in robustness against external and between nodes interference which as a result, improves the network traffic flow.

2.3.1 Introduction

There have been many proposals in multichannel communication which uses the duty cycling technique to alter the nodes sleep and listen states. The duty cycling is

an important mechanism that helps reducing the nodes energy consumption. However, adjusting the duty cycle does not solve the interference problem as external interference is unpredictable. Multi channel is a preferable solution to improved resilience against interference. However, not all channels are free from interference; thus, there is a gain to hop to another channel when the quality of the channel deteriorates. Two commonly used types of channel hopping [6] are blind channel hopping and whitelisting. In blind channel hopping, nodes choose from all available channels. Whitelisting, on the other hand, gives a set list of channels that avoids those that are known to commonly suffer interference.

Existing duty cycled multichannel MAC protocols can be categorised into two types; synchronous and asynchronous systems. These are also referred as reservation-based protocol and contention-based protocol by some authors. A synchronous system is a system that requires a tight time synchronisation between nodes. It uses time-scheduled communication where the network clock needs to be periodically synchronised to compensate for time synchronisation error in order for the nodes not to drift in time [31]. The system requires dependency on the time synchronisation and network topology. The knowledge of the network topology is required to be able to establish a schedule for the nodes to access the channel to communicate with the other nodes.

Asynchronous system on the other hand, does not require synchronisation and topology knowledge but instead is a sender or receiver initiated communication. The nodes compete to access the channel to transmit such that the node postpones its transmission if it senses that the channel is busy, by sending preamble packets, to avoid interfering with the current transmission. In asynchronous systems the nodes are able to self-configure without time synchronization and this can have advantages. There are many studies done in multichannel for both categories.

Multichannel communications have potential benefits for wireless networks that include improved resilience against external interference, reduced latency, enhanced reception rate and increased throughput. A set of existing multichannel MAC protocols are reviewed and compared, highlighting their features and limita-

tions.

2.3.2 Synchronous Systems

In synchronous systems, the multichannel MAC protocols employs time division multiple access (TDMA). It allows the channel to be divided into different time slot. TDMA-based MAC protocols allocate time slots to the nodes for data transmission or reception [31]. This helps to avoid collision between nodes during transmission as the nodes have their own time slot. However, it has a higher latency as the node has to wait to its assigned slot before it is able to transmit a packet.

TSCH [38], MC-LMAC [39] and YMAC [31] are a few examples of the existing synchronous systems. These multichannel MAC protocols are selected for review.

2.3.2.1 TSCH

The Timeslotted Channel Hopping (TSCH) [38] is MAC protocol that uses time synchronisation and channel hopping to increase reliability in the network. The nodes in TSCH are fully synchronised. The nodes are assumed to be equipped with clocks as the nodes need to maintain tight synchronisation. The clocks in different nodes could drift in time, thus the nodes need to periodically resynchronise its clock with the time-source neighbour in the absence of data to transmit. The nodes also provide their time during synchronisation to the neighbours. When the nodes have data to send, the timing information is added to the packet which simplifies the synchronisation process as the nodes are resynchronise each time they exchange data.

It is designed for optimisation, customisation and it simplifies the process of merging TSCH with protocol stack based on IPv6, 6LoWPAN and RPL. TSCH defines the mechanism to set up the schedule and control the resources allocation to each link in the network topology for execution. It also defines the mechanism that signals when a node cannot accept an incoming packet. However, it does not define when the node should stop accepting packets.

Figure 2.1 shows the TSCH schedule and terminologies used. In TSCH, time

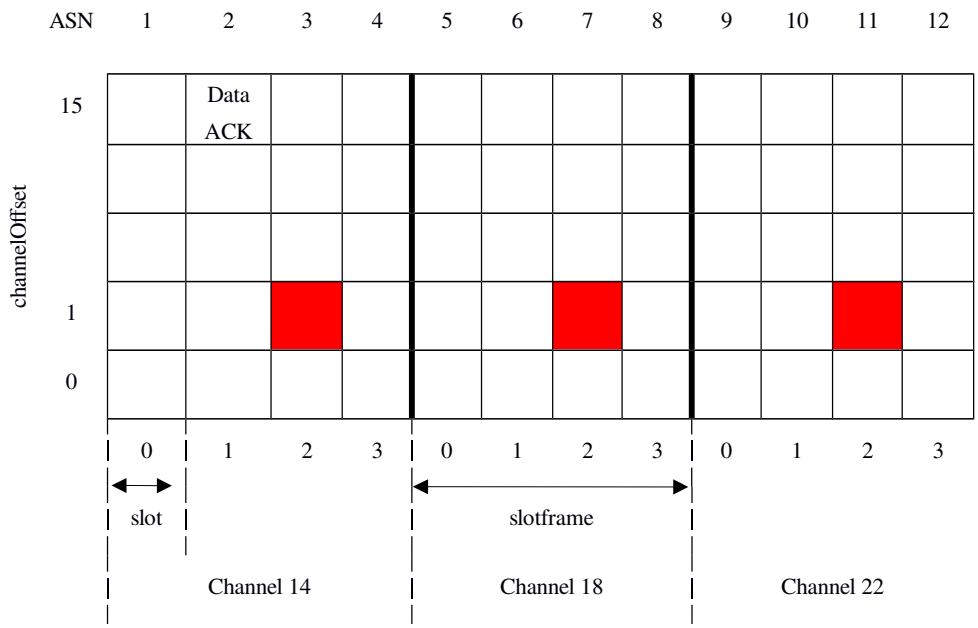


Figure 2.1: TSCH schedule

is sliced up into time slots that are appropriate for the traffic flow size. The time slot is set to be long enough to enable the sender node to send a maximum size of MAC frame to the receiver node and for the receiver to send an acknowledgement (ACK) frame to notify the sender that the frame has been successfully received.

Slotframes contain a group of time slots of equal length and priority where the slotframe repeats continuously over time. The size of the slotframe is not appointed by TSCH. Shorter slotframe has the advantage of more available bandwidth as the result of frequent repetition of the same time slot but at the cost of higher power consumption.

A single element in the TSCH schedule is called as a *cell*. The cell can instruct the node to transmit, receive or sleep. It can also be marked as both transmitting and receiving. However, transmission takes precedence over reception. The TSCH schedule also indicates the channel and address of the node for communication. The channel in TSCH is referred as *channelOffset*, which is the row in the TSCH slotframe. In a transmit cell, the outgoing buffer is checked for a packet that matches the scheduled neighbour for that time slot. Similarly, in a receive cell, the node

listens during the reserve cell for possible incoming packets. Each scheduled cell is dedicated for the node. However, a cell can be shared where multiple nodes can transmit on the same frequency at the same time. TSCH defines a backoff algorithm to avoid transmissions from nodes in the shared cells from congesting the network.

Absolute Slot Number (ASN) is a timeslot counter in TSCH that calculates the communication frequency for the sender and receiver nodes. The calculation from ASN and *channelOffset* is translated into a different frequency at different slotframe cycles. The ASN value changes at the next iteration which results in a different frequency computed for the cycle. This results in *channel hopping* where the pairs of neighbours hop between different channels at each iteration.

The advantage of channel hopping is to have retransmission on a different channel than it was transmitted previously. The new channel is likely to be a more stable link. Otherwise, it will hop to another channel on the next cycle. This increases the likelihood of succeeding than retransmitting on the same channel, thus, forming a more stable topology. Nodes on different channels are allowed to run simultaneously on the same time slot as it does not interfere with each other transmissions. Channel hopping technique helps to combat external interference and impact the nodes differently on the channel.

2.3.2.2 MC-LMAC

Multi-Channel Lightweight Medium Access Control (MC-LMAC) is a fully distributed schedule-based multi channel MAC protocol that is based on a single channel LMAC [40]. The nodes periodically use a timeslot to schedule the transmission to avoid contention. MC-LMAC does not require a centralised scheduler for timeslot allocations; instead, it is done locally by the nodes by exchanging information of their slots and channels with the local neighbours.

In MC-LMAC, the timeslots are selected with channels. The node can use the same timeslot used by a two hop neighbour if it is on a different frequency. However the node cannot use the timeslots on any frequencies that is used by the neighbours. The timeslot is selected autonomously. The node uses the same timeslot in the next frames if it does not conflict with the other nodes transmission in that slot.

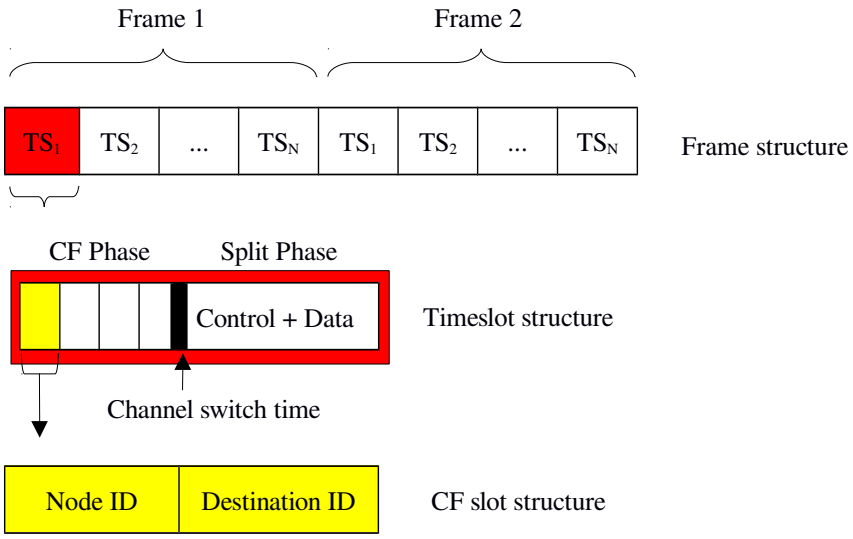


Figure 2.2: MC-LMAC scheduling

Otherwise, a new time slot is selected. The timeslot list is called *occupied slot vector* where it stores the information about the neighbours' occupied slots. The slot vector is per channel where the node can selects a timeslot for each channel given that the timeslot is free. The occupied slot vector is transmitted during the node's timeslot to the potential transmitters. All nodes are given the opportunity to select an empty slot for transmission.

A timeslot consists of a *common frequency phase* (CF) and a *split phase* as shown in Figure 2.2. All nodes listen on the common channel at the beginning of each timeslot in the CF phase to exchange control information with the neighbours. The common channel can be used for data transmission. The control information consists of the node's id and the intended destination id. The receivers listen during the whole CF phase. If it is the intended destination, the node switches to the sender's channel, otherwise it goes into the passive state. MC-LMAC uses the CF slot number as the senders channel number to avoid sending an extra transmission to the destination node.

Nodes can send broadcast messages by transmitting a broadcast address during the CF slot where the receivers switch to the sender's channel. A dedicated broadcast channel is not required. However, the CF duration increases when more

channels are used which resulted in longer listening period thus energy to wait for potential incoming packets.

The senders and the intended receivers switch to the channel where the control message and data transmission will take place in the split phase. The sender sends a control message in the form of preamble packets before proceeding with transmitting the data message. The control message that is transmitted in the split phase includes the occupied slots list. The node also sends the current slot and slot numbers in the control message prior to data transmission to detect synchronisation error by compare the slot and frame numbers that it receives in the control message with its slot and slot number. In MC-LMAC, the synchronisation is done by synchronising nodes near to the sink with the sinks and continues hop by hop where the nodes synchronise with the parents.

2.3.2.3 YMAC

Y-MAC [31] proposed a multi channel MAC protocol that uses a light weight channel hopping mechanism. In Y-MAC, time is divided into several fixed-length frames. Each frame consists of a broadcast and unicast period. The broadcast traffic is separated from the unicast traffic for a more reliable broadcast where they do not share the same queue. Figure 2.3 shows the Y-MAC scheduling. At the start of the broadcast period, all nodes must wake up to exchange broadcast messages. The nodes switch to the base channel to transmit or receive the broadcast message. Broadcast messages are only exchanged during the broadcast period. The nodes turn the radio off if there is no incoming broadcast message. The nodes will wake up again during the unicast traffic time slot. Y-MAC exploits multi channel for unicast to reduce the packet delivery latency while using a single channel which is the base channel for broadcast messages.

Y-MAC is a receiver based scheduling where the node checks the channel for incoming packet in its receive time slot. The time slot length is defined to be long enough to receive one message. The potential senders have to compete to be able to transmit. However, only the contention winner can transmit the packet to the receiver. The sender node sends a preamble until the end of the contention win-

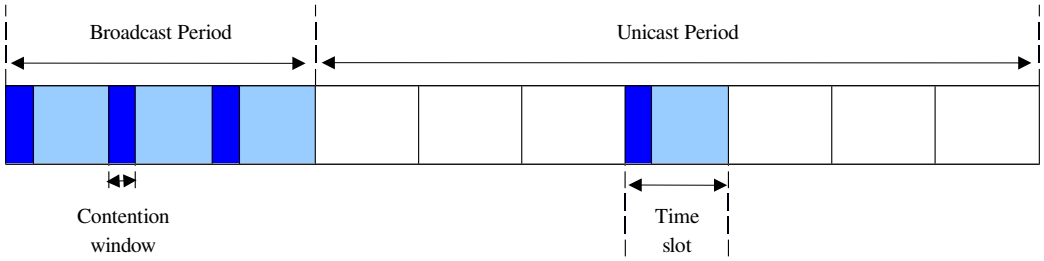


Figure 2.3: Y-MAC scheduling

dow if the channel is clear to withhold competing transmissions. At the end of the contention window, the receiver wakes up to receive the data. The receiver node transmit a small and independent packet at the start of the time slot to notify the potential senders if it waits in the next time slot for the senders to retry.

The receiver initially starts the hopping sequence on the base channel to receive the data. The receiver and potential senders hop to another available channel according to the hopping sequence to receive the following packet. The potential senders that have pending messages for the receiver will hop to the next channel and compete to transmit. The burst of messages ripple across channels which means that only one node uses the base channel at a time. This guarantees that each node will receive a packet on the base channel before it hops to another channel. The hopping sequence ensures that for any particular channel, there is only one node among one hop neighbour.

In Y-MAC, the nodes exchange the time remaining to the start of the next frame period. This information is included in the control message that is sent periodically as a broadcast to minimise the control overhead for time synchronisation. The receiving node that receives the time synchronisation information adjusts the expiration time of its timer by averaging the time remaining to compensate any timing error so that the start time to the next frame period is shorter. Time synchronisation is an important aspect in ensuring the network connectivity for communications.

2.3.3 Asynchronous Systems

Recent asynchronous multi channel MAC protocols are Chrysso and MiCMAC. These protocols are using Contiki operating system. MiCMAC is built based on

ContikiMAC, the default radio duty cycling in Contiki 2.7 that works in a single channel. The details of these are explained below. The single channel ContikiMAC is also explained as the duty cycling mechanism in ContikiMAC is important in MCRP, the protocol proposed in this thesis.

2.3.3.1 ContikiMAC

ContikiMAC [41] is the default radio duty cycling mechanism in Contiki 2.7. It is an asynchronous system where it does not need scheduling, signalling messages, and additional packet headers. It uses periodical wake ups to listen to incoming packets from the neighbours. Periodical wake ups has been used by many protocols such as B-MAC [42], X-MAC [43] and BoX-MAC [44]. ContikiMAC default wake up frequency is set to 8 Hz which results in a wake up interval of 125 ms. Frequent wake up would enable quicker packet detection in the case of frequent packet transmissions at the cost of higher network power consumption.

The receiver is kept on when it detects a packet transmission during a wake up. ContikiMAC wake ups use an inexpensive *Clear Channel Assessment* (CCA) that relies on the threshold of the *Received Signal Strength Indicator* (RSSI) to signify the radio activity on the channel. A positive CCA represent a clear channel if the RSSI is below a threshold and the CCA returns a negative value if the channel is currently in use. The ContikiMAC CCAs do not detect packet transmission. They are used to detect the activities on the radio signal which could be that (i) a neighbour is transmitting to a receiver, (ii) a neighbour is transmitting to other receivers, or (iii) other devices that radiate radio energy.

ContikiMAC uses a fast sleep optimisation to enable the receivers to quickly go to sleep in the case of spurious radio interference that is a false positive wake ups. The receivers can go back to sleep if (i) the duration of the radio activity is longer than the maximum packet length, (ii) the silence period is longer than the interval between two successive transmissions or (iii) the start of packet is not detected.

Another feature of ContikiMAC is the transmission phase-lock mechanism. This feature has been suggested by WiseMAC [45] previously and has been used by other protocols. In Contiki, the phase-lock optimisation is implemented as a

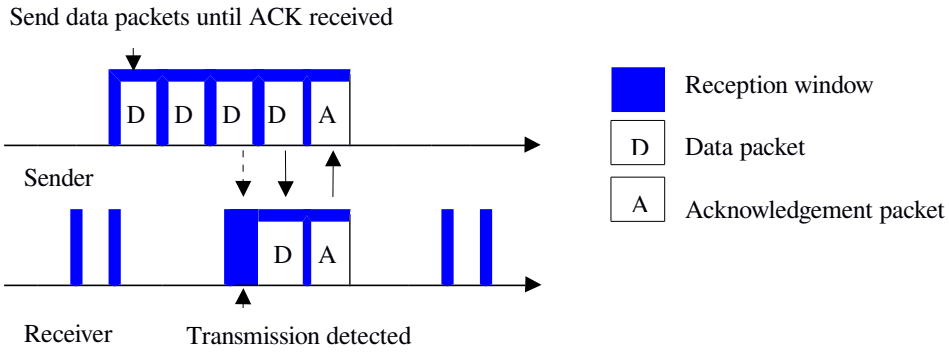


Figure 2.4: ContikiMAC unicast transmission

separate module from ContikiMAC. It manages a list of neighbours and their wake up phases.

When a sender has a packet to send, the sender repeatedly sends the packet until it receives a link layer acknowledgement from the receiver to indicate that the packet has been successfully received. Figure 2.4 shows ContikiMAC unicast transmission. A sender can learn the receiver's wake up phase through the link layer acknowledgement. This reduces the number of transmissions required significantly as the sender can send the packet shortly before the receiver is expected to be awake which as a result, reduces the network congestion.

However, a broadcast packet does not result in a link layer acknowledgement. The packet is instead repeatedly sent in the full wake up interval to reach all neighbours. During broadcast, the sender can turn the radio off between each packet transmission to save power as it does not expect to receive any link layer acknowledgement.

2.3.3.2 MiCMAC

MiCMAC [8] is a distributed channel hopping variant of ContikiMAC. It inherits ContikiMAC basic design and further extended to support multi channels. It also extends ContikiMAC's phase lock to include *channel lock* for wake up channel. MiCMAC is independent from the other layers in the protocol stack. It is compatible to run with RPL routing.

In MiCMAC, the node switches to different channel each time it wakes up. The channel is generated using a Linear Congruential Generator (LCG) for a pseudo-random sequence. The generated sequences are random and use each possible number within the range once before the sequence is repeated. MiCMAC uses a predefined hopping sequence that is provided in a static table of all sequences instead of generating the hopping sequences at runtime to increase optimisation. The sequence is selected according to the node's MAC address.

When communication with the neighbour for the first time, the sender transmits strobes repeatedly on a channel for a maximum number of channels wake up. The sender picks any channel. The receiver wakes up on different channel each time following the pseudo-random sequence and it will wake up exactly once on the sender's strobing channel. The receiver sends an ACK that includes the node's pseudo-random generator parameters. The sender will use this information and the number of periods elapsed since the last successful unicast with the receiver to generate the next wake up channels for the node.

The sender switches to the receiver's expected channel just before it wakes up, checks the radio activity using CCA and sends the packet if the channel is clear. The receiver will reply with an ACK to the sender. The sender updates its information of the receiver's wake up time and channel before it goes back to sleep. If the ACK is not received, the sender assumes that the receiver's wake up time and channel is wrong. The sender updates the receiver's information.

MiCMAC supports broadcast and introduces two variants of MiCMAC; MiCMAC and MiCMAC-BC. In the basic MiCMAC broadcast, only one of the available channels in the sequence is strobes continuously for a maximum number of channels wake up. It has the disadvantage of increased energy and channel used from the broadcast message. MiCMAC-BC on the other hand uses a dedicated broadcast channel. The nodes wake up on the broadcast channel at every period in addition to the unicast pseudo-random wake up channel. However, it has the disadvantage of reduced robustness as the broadcast channel is fixed and MiCMAC requires two wake ups at every period which are for the broadcast, followed by the unicast chan-

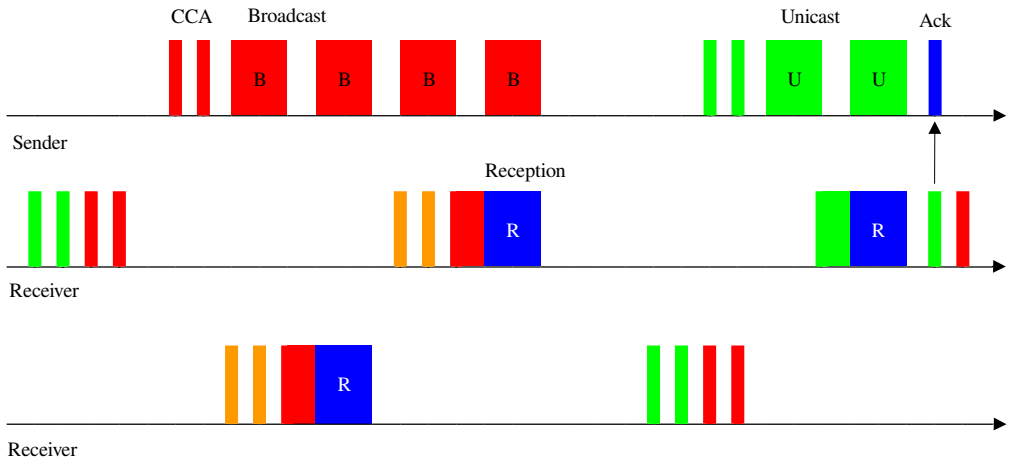


Figure 2.5: MiCMAC unicast and MiCMAC-BC transmissions

nels. Figure 2.5 shows MiCMAC-BC and unicast transmissions.

In MiCMAC, the performance degraded when it considers all 16 channels as it includes the high interference channels for transmissions. It also increases the broadcast and channel-lock costs. MiCMAC showed an optimal performance with 4 channels.

2.3.3.3 Chrysso

Chrysso [7] is a multi channel protocol extension that is specifically tailored for data collection applications. Chrysso is implemented in Contiki 2.4 using *collect* routing protocol which is a data collection. Chrysso switches the channel of the nodes that are affected by the external interference to a new set of channels when the interference is detected to evade the interference source.

Chrysso allows the parent to coordinate the channel switch when interference is detected for each individual parent-children group. Chrysso operates on two channels; one for incoming (called as inchannel) and the other channel is for outgoing (outchannel) traffic. Chrysso maintains a pre-defined logical list of available channel. The parent and children use this list to ensure consistency when they are switching to the next channel. Chrysso implemented five channels which are channel 26, 14, 20, 11 and 22 and evaluated Chrysso's performance using these channels in this particular order.

Chrysso consists of a set of control loops; the inner and outer loops, that decides to maintain or switch the parent-children's channel. The inner loop is responsible for the parent-children channel switching coordination when there is external interference. The child node collects data from the channel quality monitor periodically which is then included onto the data packet. The parent uses the average over congestion backoffs values to measure and determine if there is external interference in the current measured channel quality. If the computed value exceeds a predetermined threshold, the children are notified of the channel switching request before the parent switches to the next inchannel.

Chrysso uses the outer loop during severe interference that blocks any communication where the parent-children channel switching coordination could not be invoked. The outer loop is a watchdog mechanism where autonomous channel switching is initiated when the inner loop could not be triggered. The nodes, both parent and children independently switch to the next channel in outer loop. The child node monitors the number of failed transmissions. If it exceeds a predefined threshold, the outchannel is switches as it indicates that the channel has severe interference. Likewise, the parent records the number of packets it received and switches to the next inchannel instantly and autonomously if the received packets are below a pre-set threshold value of the expected packets.

The watchdog initiates the *scan mode* to find a new parent when the nodes have lost contact with the parent after the channel switches. The node scans through all available channels except the previously used outchannel to find a new parent as neighbours are now operating on different channels. The scan mode is triggered on demand as it uses additional overhead for processing and consumes energy.

2.3.4 Comparison and Discussion

The reviewed MAC protocols features are summarised in Table 2.1. Several issues that exist are in term of:

1. **Synchronous versus asynchronous design** - Both designs have their advantages and disadvantages. However, synchronous MAC protocols require the network topology to be known before it can schedule timeslots to avoid con-

Protocol	Medium Access	Channel Assignment	Channel Switching	Common Period	Broadcast
TSCH	TDMA + collision window	Dynamic	Once per cycle	No	Yes
MC-LMAC	TDMA	Senders	Once per time slot	Yes	Yes
Y-MAC	TDMA + collision window	Dynamic	Once per time slot	Yes	Yes
MiCMAC	MiCMAC	Dynamic	One per wake up time	Yes	Yes
Chrysso	Contiki-MAC	Dynamic	When require	No	No

Table 2.1: Comparison of studied MAC protocols

flict between the nodes. Asynchronous MAC protocols depend on the channel condition and the nodes compete for the channel access.

2. **Sender versus receiver initiated design** - Most MAC protocols presented are both sender and receiver initiated where the channel decisions depended on both nodes.

3. **Channel hopping design** - The channel quality checking at run time costs a lot of energy. In most of the MAC protocols studied evaluation, the protocols are tested using a few selected channels instead of considering all available channels. This limits the spectrum usage of the frequencies.

4. **Broadcast support** - Most MAC protocols presented provide a broadcast support except for Chrysso. The protocols specified a broadcast period at every wake up on a fixed channel which reduces the robustness of the protocol as broadcast will occurs on the same channel each time. TSCH on the other hand treats broadcast message the same as a unicast message where it needs to select a slot before it can proceed but with a broadcast MAC address as the destination.

Based on these reviews, the MAC protocols have many features that ease the multi channel processes. However, to further improve multi channel protocols, cross layers decisions are required. This could lighten the processing load at the MAC protocol and to allow switching decisions to be determined at run time on the upper layer which could compute better decisions.

2.4 Routing Protocol

Routing protocols are responsible in routing data from the sender to the intended receiver across the network. In WSNs, the sensor nodes are restricted to a transmission range of approximately 20-30 metres indoors and 75-100 metres outdoor [46]. The routing protocols are required to manage and maintain the routes to ensure reliable communications between the limited range nodes. The routing protocols in WSNs are different than the traditional routing protocols as sensor nodes have limited processing capabilities, limited storage and use different operating system.

2.4.1 Introduction

In WSNs, flooding and gossiping are two classical approaches to relay data to the destination [47]. Gossiping is a slightly enhanced version of flooding. These data delivery approaches does not depend on any routing protocols and network topology. In flooding, the packet is broadcasts to all the node's neighbours and the neighbours that receive the packet will continue to broadcast the packet to their neighbours until the packet destination is found or the maximum number of hops allowed is reached. In gossiping, instead of flooding all the neighbours, it selects a random neighbour to forward the packet. The node that was selected will pick another random neighbour to continue sending the packet until it arrives at the intended receiver. Flooding and gossiping approaches are easy to implement. However, the approaches cause duplicated messages due to overlapping nodes that receive and send the same packet. The approaches also consume a large amount of energy. Gossiping causes propagation delay as the nodes selected are not guarantee to be the nearest node to get to the destination node.

Routing protocol is important in ensuring reliable packet delivery in WSNs.

Several crucial criteria in the design of a routing protocol are in term of scalability, reliability, power consumption and adaptability. In WSNs, sensor nodes are typically densely deployed in the error prone wireless channels. This places the importance of scalability in the routing protocol to reduce conflict between the nodes and external interference from other devices while maintaining reachability between nodes. Sensor nodes are equipped with limited battery power. The routing protocol should consider the sensor nodes battery level before deciding on the routes. It can prolong the network lifetime by considering alternative routes in order to avoid draining lower energy nodes. Throughout the network lifetime, the nodes may fail, join or move to a different location than the nodes were initially. The routing protocol should be able to adapt to these changes and updates the routes accordingly.

Routing protocols are aimed to provides an optimised, scalable and energy efficient routes in the network which as a result, could prolong the network lifetime.

2.4.2 Classification of Routing Protocols

There have been many routing protocols that were developed for WSNs. The routing protocols can be classified into 4 types [47]; (i) flat based and data centric, (ii) location based, (iii) network flow and quality of service (QoS) aware, and (iv) hierarchical based. These classifications are explained below with examples of the existing routing protocols for each type.

2.4.2.1 Flat based and Data Centric

Data centric routing is a neighbour-to-neighbour query based routing. It uses attribute-based name or meta data to refer to a specific data. The sender sends data queries to a certain region and waits for the sensors to send a reply with the queried data. SPIN [48] and Directed Diffusion [49] are a few examples of the earlier data centric protocols.

SPIN is the first data centric protocol. It uses high-level descriptors or meta data to refer to the data. These meta data are exchanged using the data advertisement mechanism among sensors before transmission. Each node in SPIN is only required to know its immediate single hop neighbour. This allows topological

changes to take place locally. The nodes that receive the meta data will then advertise the data availability it to its neighbours. This allows interested nodes to query the data. However, SPIN does not guarantee the delivery of data to the interested node. The data delivery is decided by the nodes that are situated between the source and destination nodes. If the in between nodes are not interested in the data, the data will not be delivered to the destination.

Directed Diffusion is a query-driven data delivery model. It is an important milestone in data centric routing. Directed Diffusion aims at diffusing data using attribute-value schemes. The data on the sensor is queried in an on demand basis using the attribute-value pairs. An interest or task is defined using a list of attribute-value pairs to create a query. The sink broadcasts the interest through its neighbours. The interest is cached at the receiving nodes for later use. Caching helps to increase the routing energy efficiency and minimise delay. It is used to compare the received data with the values in the interests. The reply link to the neighbour from which the interest was received is called a gradient. The paths are established between the sink and sources based on the interest and gradient utilisation where one of the paths is selected as reinforcement. The interest is then resent by the sink using the selected path with a smaller interval. Directed Diffusion selects a new or alternative path that sends data in lower rates when the current path fails.

In Directed Diffusion, the sink queries the sensor nodes for the specific data. SPIN however, advertise the available data to allow interested nodes to query the data. There are many other protocols that have been proposed either based on Directed Diffusion such as Rumor Routing [50], GBR [51], CADR [52]) or following a similar concept such as TEEN [53] which is also a hierarchical-based, and ACQUIRE [54].

2.4.2.2 Location Based

In location based routing, the sensor nodes locations are used to estimate the energy consumption between the distances of the two known nodes. As the location of the nodes is known, the number of transmissions required can be reduced as the transmissions can now be targeted to the specific region. However, the nodes are

required to be equipped with Global Positioning System (GPS) to allow location of the nodes to be detected. GEAR [55] and GAF [28] are two examples of location based routing.

Geographic and Energy-Aware Routing (GEAR) is an energy efficient routing protocol that are used to queries the targeted regions. The sensor nodes are equipped with GPS to enable location detection. The idea is to restrict the number of transmissions by only considering a certain region rather than to the whole network. The nodes are location and energy aware. The nodes are also aware of the neighbour's residual energy. Each node keeps an estimated cost, which is the residual energy and distance to the destination. This enables GEAR to use this information to select the nodes to route a packet to the destination region efficiently. A node sends a packet to the target region. When the node receives a packet, it checks if there is a neighbour that is closer to the target region than itself. The nearest neighbour to the target region is selected. If the packet has reached the region targeted, the packet can be diffused by recursive geographic forwarding or restricted flooding to reach all nodes in the region. In recursive geographic forwarding, the region is divided into four sub regions. The packets are made into four copies and forwarded into the regions.

Geographic Adaptive Fidelity (GAF) is another energy-aware location based routing. It turns off unnecessary nodes in the network without affecting the routing coverage to conserve energy. GAF forms a virtual grid for the covered area where nodes in the same grid are considered to have equivalent cost for routing. The nodes are grouped into the virtual grid according to the nodes location indicated by the GPS. Some of the nodes in the same virtual grid turn off their radio to save energy and wake up before the currently active nodes expire and go to sleep. GAF keeps a representative node awake on each virtual grid for routing. GAF increases the network lifetime as it exploits the location of the nodes in order to minimise the number of awake nodes in each grid to conserve energy.

2.4.2.3 Network Flow and QoS-aware

In network flow and QoS aware routing, the routes setup take into account the network flow problems and the quality of service requirements such as the end to end delay, routes reliability and fault tolerance in routing. SAR [56] and maximum lifetime energy routing [57] are examples in network flow and QoS aware routing. These routing protocols try to find a balance between energy consumption and QoS requirements.

Maximum lifetime energy routing solution has the objective of maximising the network lifetime by considering the nodes remaining energy to define the link cost for transmission using the link. The protocol uses Bellman-Ford shortest path algorithm to compute the least cost paths to the destination.

Sequential Assignment Routing (SAR) is the first protocol that considers the QoS in its routing decisions. The path is selected based on the energy resources, QoS on each path and the packet priority level. SAR proved to consume less energy when it considers the packet priority than other minimum-energy routing that only focuses on the energy consumption. SAR tries to minimise the average weighted QoS metric throughout the network lifetime. SAR maintains multiple path form nodes to the sink to ensure fault tolerance and easy recovery. However, this resulted in high overhead to maintain the paths in a large network.

2.4.2.4 Hierarchical

Hierarchical based routing aims to scale a large set of sensor nodes that cover a wider area of interest by enabling multi hop communication while maintaining efficient energy consumption. A single based routing is not scalable and causes the network to overload when the number of nodes increases which resulted in conflict in transmissions, thus congestion. Hierarchical based routing usually group nodes into clusters and performs data aggregation and fusion to eliminate duplicate and reduce the number of transmitted messages. LEACH [26] is one of the first hierarchical routing in WSNs. The idea proposed in LEACH has inspired many hierarchical routing protocols such as TEEN [53], APTEEN [58], PEGASIS [59], Hierarchical-PEGASIS [60] and HEED [61].

Low-energy Adaptive Clustering Hierarchy (LEACH) is one of the most popular hierarchical routing protocols in WSNs. LEACH forms dynamic clustering of nodes based on the received signal strength and selects a node as the local cluster head. The cluster head is used for data processing such as data aggregation and fusion of the nodes within the cluster and to route the processed data to the sink. It consumes a larger amount of energy for data processing. The cluster head changes periodically where the node is selected by random to become the cluster head. This is done in order to balance the energy dissipation of nodes which increases the network lifetime. The random selection ensures that the nodes die randomly to avoid the network from not functioning. However, it brings extra overhead when the cluster head changes as it has to advertise to the nearby nodes of the changes. LEACH is a distributed routing and does not require global knowledge of the network. It uses a single hop routing where the node transmits directly to the cluster head, and the cluster head to the sink. LEACH is not applicable to large network that requires multi hop.

Contiki provides support for two hierarchical based routing protocols, Contiki Collect protocol which is the Contiki implementation of Collection Tree Protocol (CTP) [62, 63] and RPL [64]. Chrysso uses Contiki Collect protocol as the routing protocol and RPL is compatible with MiCMAC and it is the main routing protocol in MCRP.

Contiki Collect protocol and CTP are data collection protocols that are address free. The nodes send the data usually towards the sink without specifying the node's address. The routing protocol builds a tree originating from the tree and the nodes send periodic announcements which contain the number of hops away from the sink. Both protocols use the expected number of transmissions (ETX) as the metric to find the paths that requires the minimum number of transmission to reach to the root. The nodes start sending messages towards to root once the tree is built. The messages are sent using hop-by-hop reliable unicast. Contiki Collect protocol and CTP are not IPv6-based. Contiki Collect protocol uses Contiki Rime stack which is Contiki's lightweight communication stack explained in Section 4.1.1.

RPL is designed largely based on CTP. RPL is a distance vector routing protocol that uses IPv6. It is a Destination Oriented Directed Acyclic Graph (DODAG) that is routed at a single destination, which is the root. RPL supports any-to-any routing where the traffic is routed upwards until a common ancestor of the destination and source is found, then downwards to the destination. RPL uses a simple rooted topology instead of a full mesh. It maintains reliable paths to a single destination which allows RPL to scale to large networks while keeping the routing overhead to a minimum at the cost of increase hop count where the nodes traffics are routed upwards until a common ancestor is found.

RPL is constructed using an Objective Function (OF) that specifies the routing metric, routing constrains and other functions to construct the topology. The OF is application dependant as RPL does not define any specific OF. There are two OF that are provided by RPL which are a simple hop count [65], where it selects the path that has the shorter path, and ETX [33], that depends on the path that requires less transmissions [66, 67, 68].

The distance from the node relative to the other nodes with respect to the root is called *rank*. The rank increases away from the root and decreases when it is nearer to the root. Rank is used to avoid routing loop in the topology as the node's position relative to the other nodes is known. RPL has rank hysteresis mechanism to avoid frequent parent switching in the case of little rank improvements.

RPL has four types of ICMPv6 control messages that are used for topology maintenance and information exchange which are DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS) and an optional DAO-ACK message [64]. DIO is the main routing control information that includes the node current rank, configuration parameters and the root IPv6 address. DIS is used to enable a node to enquire DIO messages from a reachable neighbour. DAO is used to propagate destination information upwards along the DODAG. It also enabled down traffic to be supported. DAO-ACK is used as a DAO message response to acknowledge the DAO message by the DAO recipient.

The topology is constructed from the root node. It sends DIO messages to the

Protocol	Scalability	Route Metric	Periodic Message Type	Robust
SPIN	Good	Single hop neighbour	Advertise to all neighbours	Robust
Directed Diffusion	Good	Best path	Query messages	Low
GEAR	Limited	Best route	Hello messages	Good
GAF	Limited	Shortest path	Discovery messages	Good
SAR	Limited	Path with minimum average weighted QoS metric	Hello messages	Low
LEACH	Good	Shortest path	None	Good
CTP	Good	ETX	Beacons	Good
RPL	Good	ETX	DIO messages	Good

Table 2.2: Comparison of studied routing protocols

reachable nodes. The nodes that receive the message run an algorithm specified by the OF to select a parent. The nodes compute their rank and send an update in the DIO message to the neighbours.

RPL uses *Trickle* algorithm [69]. Trickle is used to control the message sending rate. In RPL, Trickle reduces the control messages rate by exponential increase to avoid the control messages from congesting the network. DIOs are sent periodically where the duration is doubled each time a DIO is sent until it reaches Trickle maximum possible value.

2.4.3 Comparison and Discussion

The routing protocols reviewed are summarised in Table 2.2. Based on these, the important factors [32] that influence the routing protocols are:

1. **Node deployment** - Sensor nodes are scattered randomly and require scalable protocol to allow routes to be formed. In a large region, the nodes require multi hop to reach to the root node.

2. **Data delivery models** - Depending on the application, the data can be continuous, event-driven, query-driven or hybrid. The nodes send periodic data in continuous-driven and in event and query-driven, it depends on when the

data is triggered and query is generated.

3. **Energy** - The distance from the node to the sink influence the energy consumption of the path. In many routing protocol, the shortest path is used as an indicator to represent energy efficiency.

Chapter 3

Multichannel Cross-Layer Routing Protocol

WSNs often suffer from frequent occurrences of external interference such as Wi-Fi and Bluetooth. Multichannel communications in wireless networks can alleviate the effects of interference to enable WSNs to operate reliably in the presence of such interference. As a result, multichannel solution can improve the network efficiency of spectrum usage, network stability, link reliability, minimise latency and minimise the number of packet loss, hence, retransmission.

This chapter presents Multichannel Cross-Layer Routing Protocol (MCRP), a decentralised cross-layer protocol with a centralised controller. Our cross layer multichannel protocol focuses on the network and application layers. This allows channel assignment decisions to be made thoroughly without being limited by the low layer complexity. The system has two parts: a central algorithm which is typically run by the LPBR and selects which channel each node should listen on; and a protocol which allows the network to communicate the channel change decision, probe the new channel and either communicate the success of the change or fall back to the previous channel. MCRP concentrates on finding channels for the nodes that are free from or have low interference. It allows the allocation of these channels in a way likely to minimise the chances of nodes which are physically near to communicate on the same channel. Hence, it reduces cross interference between different pairs of nodes.

3.1 MCRP Design

The design of the multichannel protocol is based on several crucial observations:

i. **Channel assignment** - Sensors have limited memory and battery capabilities. In order to maximise the sensors lifetime, a centralised LPBR that has larger memory and fully powered is used for decision making. LPBR has complete knowledge of the topology which enables it to make good channel assignment decisions based on a two-hop colouring algorithm.

ii. **Interference** - External interference cannot be predicted, thus channels cannot be allocated beforehand as it varies over time and locations. It is impossible to determine a single channel that is free from interference at any location. The protocol checks the channel condition each time before deciding on a channel change to reduce interference and maximise throughput.

iii. **Frequency diversity** - Multichannel increases the robustness of the network towards interference. However, applying multichannel to the existing RPL may hinder detection of the new nodes and cause problems for maintaining the RPL topology. Two mechanisms are introduced to overcome this problem. These solutions are explained in details in Chapter 4. Existing nodes maintain a table of the channels on which their neighbours listen and use unicast to contact those nodes. New nodes listen on a Contiki default channel (26) and when connecting search through all channels. As in RPL, periodically all nodes broadcast RPL control messages on the default channel in an attempt to contact new nodes.

MCRP is build based on these observations to overcome the shortcomings of sensors and sensor networks. The rest of this chapter focuses on MCRP designs in channel selection strategy, channel switching decisions, channel quality checking and the reconnection strategy.

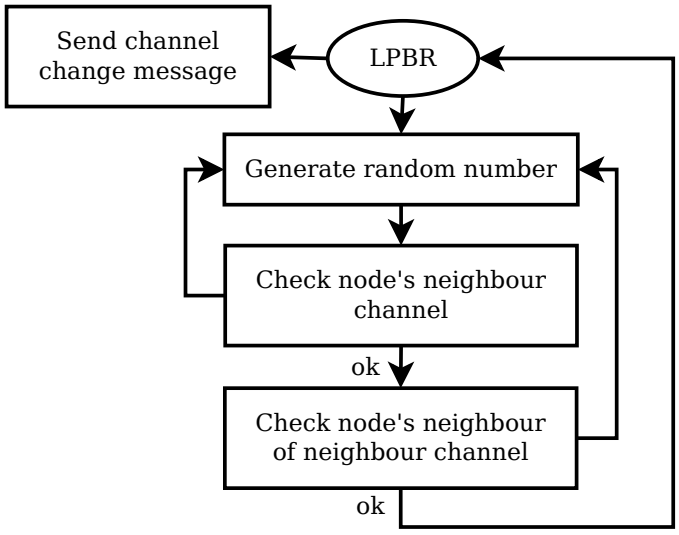


Figure 3.1: Channel selection strategy

3.2 Channel Selection Strategy

One main advantage of the proposed system is generality. Any algorithm can be used at the LPBR to assign channels. MCRP uses a two-hop colouring algorithm to select a channel to be assigned to a node. The two-hop colouring algorithm attempts to ensure that nearby nodes do not communicate on the same channel and risk interfering with each other. The protocol is inspired by the graph colouring problems [70]. The core idea is that no node should use the same listening channel as a neighbour or a neighbour of a neighbour (two hops). This allows fair load balancing on the channels and reduces channel interference that could occur when two nearby nodes transmit together on the same channel. The nodes used in this for this experiment have a transmission range of approximately 20-30 metres indoors and 75-100 metres outdoors [46]. It could be the case that many nodes in a sensor network are in the transmission range of each other and potentially interfered with.

All nodes are initialised to channel 26 which is the common default channel for Contiki MAC layer since it often has fewer interference problems with Wi-Fi and other sources. The studies in [7, 8, 6] use a set list of whitelisted channels in their experiments and have channel 26 in common. The usual RPL set up mechanism is used to exchange control messages that are required to form an optimised topology

before channel assignments can take place. The nodes will only be on the same channel once during the initial setup. This enables the node to detect and find nearby neighbours that are in range before it can decides on the best route based on the list of neighbours it can be connected to.

In the two-hop colouring algorithm, the LPBR chooses a node to which it will assign a channel to listen on. The selection is random (from channels 11 to 26) based on the full range available [1]. The channels that were tested to have severe interference for one node might gives good result for another node depending on the location of the node which might not be within the range of where the channel has severe interference previously. MCRP has its channel quality checking mechanism before it decides on a channel which allows random channel selection to take place.

The protocol checks neighbours and neighbours of neighbours to see if any of those are listening on this channel already. If any are, a new channel is picked from the remaining list of available channels. If the LPBR has knowledge of existing bad channels then those channels can be blacklisted. Knowledge of channel interference which is gained by probing can be used to decide that a channel should not be used. If a channel is found then the channel switching protocol is triggered. If no channel can be found meeting these conditions, the current channel is kept. Figure 3.1 summarised the strategy in LPBR channel selection.

The node selection algorithm must only attempt one channel change at a time to ensure probing is done on the correct new channel and for the node to finalise the channel to be used before another node attempts a channel change. The protocol ascertains that the channel change attempt will always result in a message returned to the LPBR either confirming the new channel or announcing a reversion to the old channel. Until one or other of these happens, no new channel change will be made to enable the neighbours transmitting on the correct channel.

3.3 Channel Switching

Figure 3.2 shows the state machine for the channel switching protocol. As explained in the previous section, a choice of a new channel by the channel selection protocol

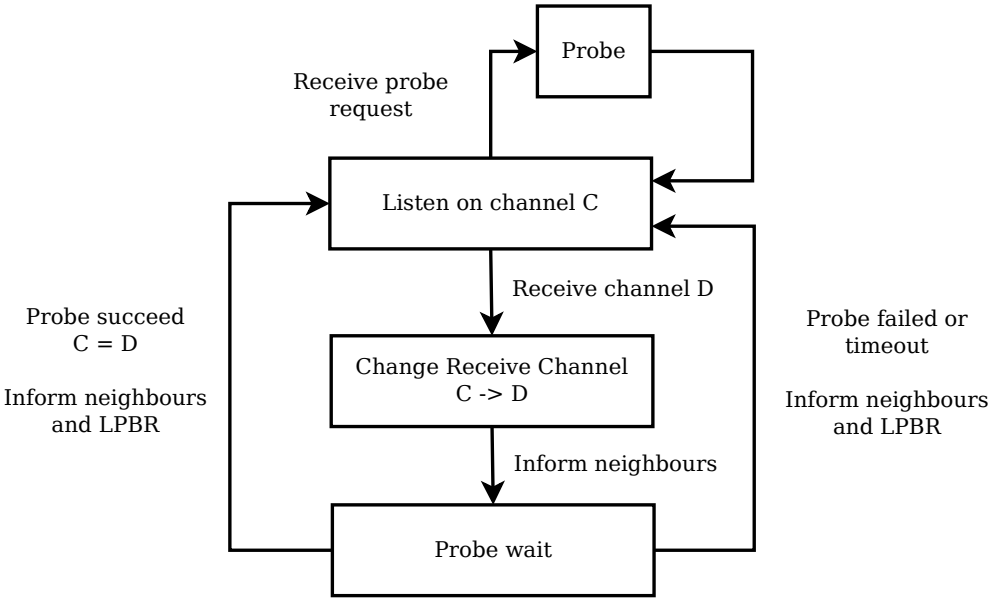


Figure 3.2: Channel switching processes

causes a change channel message to be sent to the appropriate node. Upon receiving a channel change message, a node *N* stores its current channel *C* and communicates to all its neighbours the new channel *D* that it wishes to change to. Those neighbours will update their neighbour tables to ensure that they now send to node *N* on channel *D*. The node *N* begins the channel quality checking process with each neighbour in turn by sending them a probe request. If this process fails for any neighbour then the node reverts to channel *C*. If all channel quality checks succeed, the node *N* will listens on channel *D*. In both cases, node *N* informs its neighbours of the decision to channel *C* or *D* and informs the LPBR of the channel checking results. The channel checking process uses probe packets that might interfere with other transmissions temporarily. However, it is important to emphasise that the network remains fully functional and connected at all stages of this protocol.

3.4 Channel Quality Checking

The channel quality checking is invoked each time a node changes channel after receiving a message from the LPBR. A node *N* changing to channel *D* informs all neighbours in turn, of the new channel *D* it will be listening on as described

in the previous section. It then enters the *Probe Wait* state and begins channel quality checking with each tree neighbour in turn. In describing the channel quality checking process, it is worth emphasising the distinction between neighbours and tree neighbours. Node neighbours are all nodes that a given node knows it could transmit to. Tree neighbours are the nodes that a node does transmit to through the topology formed by the RPL protocol.

In the *Probe Wait* state, node N sends a *Probe* message to each neighbour in turn. The neighbours respond to the message by sending eight packets to N on the new channel D. The buffer can accommodate eight packets at a time. As the packets might not be sent immediately due to wakes up and collisions, sending more packets would have the risk of being dropped. The condition of the channel is further investigated through the number of retransmissions and packet collisions of the probing packets for accuracy of the channel condition.

If the probing process times out (because of some communication failure) or the number of probe packets received is above a threshold (currently set to 16, including retransmissions and collisions) then node N immediately exits *Probe Wait* state and reverts to channel C its previous channel.

All neighbours are informed of the change back to channel C and the LPBR is informed of the quality check failure with a summary of all probes received. If, on the other hand, all channel quality checks succeed, the change to channel D becomes permanent for node N and it informs the LPBR of the results of the probing (numbers of packets received) and the channel change.

Probing is essential to make the channel change decision. It gives a quick overview of the channel condition based on the number of probing messages received. It is worth noting that probing is only done between the node and the tree neighbours. Neighbours that are not tree neighbours will not use the node as a route during their transmission thus, there is no need for probing to take place with those neighbours. However, the neighbours still need to know the channel value given that RPL control messages are sent to neighbours directly without using the routes.

3.5 Reconnection Strategy

RPL topology stability (using routing metric) remains the same in multi channel [66, 64]. The nodes can still change the parents as usual as all neighbours know each other new channels. The neighbours that are not part of the route do not probe the parent when making the channel decision. However, the neighbours are informed of any channel changes.

This enables the topology to be optimised when communication fails and further improved through MCRP as the nodes have knowledge of the listening channels of all other nodes within the range. If a new node tries to join the topology, it sends a RPL control message through all channels as the listening nodes are unlikely to be on the default channel.

The listening nodes send a broadcast on a default channel to discover new nodes (in Contiki default, new nodes will start on channel 26) and send RPL messages through unicast when the neighbours are known to reduce unnecessary transmissions in broadcast. New nodes and nodes which fall off the network can now rejoin on many potential channels.

Chapter 4

Implementation

MCRP is implemented on the TelosB mote platform. It uses Contiki, a lightweight operating system as the software development platform that supports the standard IPv6. The implementations of MCRP across the layers in Contiki are described in details, specifying the changes that were introduced and undertaken in addition to the default parameters and settings in Contiki.

4.1 Contiki

Contiki is defined by four layers network stack: the network layer, the MAC layer, the radio duty cycling (RDC) layer and the radio layers. The network layer includes support for TCP, UDP, IPv6, IPv4, RPL routing protocol and 6LoWPAN. IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) is a header compression and fragmentation format for IPv6 packets delivery over IEEE 802.15.4 networks [71]. Contiki implements the minimal set of IPv6 protocol required, 6LoWPAN

Contiki	IoT/IP	Applications
Network	Application	HTTP
	Transport	TCP, UDP
	Network, Routing	IPv6, IPv4, RPL
	Adaptation	6LoWPAN
MAC	MAC	CSMA/CA
RDC	Duty Cycling	ContikiMAC
Radio	Radio	IEEE 802.15.4

Table 4.1: Contiki network stack

adaptation layer for IPv6 header compression and fragmentation which is routed over the low power and lossy networks (LLN) in RPL.

Contiki's configuration options for communications, buffer management and network interface are explained before looking into MCRP implementation for ease of reading.

4.1.1 Communication Stacks

Contiki contains two communication stacks, uIP and Rime. uIP [72] is a small RFC-compliant TCP/IP stack that is designed to contain only the essential (required/necessary) features to provide Contiki with TCP/IP networking support to allow Contiki to communicate over the Internet compared to the traditional TCP/IP that requires (high) resources to fit in a limited RAM capabilities of a sensor. The minimal set of features includes IP, ICMP, UDP and TCP protocols compared to the traditional TCP/IP that requires (high) resources that could not be supported in the limited RAM capabilities sensor. The uIP is mostly concerned with the TCP and IP protocols and upper layer protocols [73, 74].

Rime is Contiki's lightweight communication stack that aims to simplify the sensor network protocols implementation by reusing code in a layered manner [75]. Rime combines layers of simple communication abstractions to form a powerful high-level abstraction ranging from best-effort anonymous broadcast to reliable network flooding. Parts of the Rime stack can be used by the underlying MAC or link layer. Additional protocols that are not in Rime can be implemented on top of the stack.

Applications in Contiki can decide to use one of the communication stacks available, both or none at all. uIP can run over Rime and similarly, Rime can run over uIP [76].

4.1.2 Buffer Management

Chameleon [77] is a communication architecture in Contiki that consists of Rime communication stack and a set of packet transformation modules. It uses an abstract representation of the information which allows access to the low-level features of

the underlying MAC and link layer protocol from the applications and layers implemented on top of the Chameleon architecture. It also allows the output from the protocol stack to be adapted by other communication protocols. In Chameleon architecture, the parsing of its header is separated from the communication stack. This allows uIP or Rime communication stack to be used as described in Section 4.1.1. Chameleon architecture enables the layers to access information without violating the layering principle.

In buffer management module of Chameleon architecture, all incoming and outgoing packets from the applications and packet attributes are stored in a single buffer called the Rime buffer [72, 77, 75, 9]. All layers of Contiki's network stack including uIP, Rime and the underlying link layer operate on the same packet buffer for the buffer management. The Rime buffer has no locking mechanisms as it is a single priority level buffer. The buffer only holds the current packet.

Protocols that need to queue packets allocate the queue buffer dynamically. Queue buffer is used to hold the queued packets such as for MAC protocols that have high rate of incoming and outgoing messages before it can send or process the receiving packets; or when the radio is busy and the MAC protocol has to wait for the radio medium to be free before proceeding with transmissions. Queue buffer is used to avoid the risk of the packet overwritten by the newer packet. The Rime buffer contents are copied into the queue buffer when there is a queue buffer allocated.

4.1.3 Tunslip

Serial Line Internet Protocol (SLIP) [78] is a protocol that has a low complexity and small overhead commonly used to encapsulate IP packets for point-to-point communication between the sink (LPBR) and the device connected such as an embedded PC across the serial connections. The communication between the devices can take place on any reliable network such as the Ethernet where the LPBR can be connected to an embedded PC which contains an Ethernet interface.

Contiki provides support to communicate with devices using SLIP through its tunslip tool. Tunslip is used to bridge the IP traffic between the LPBR and the

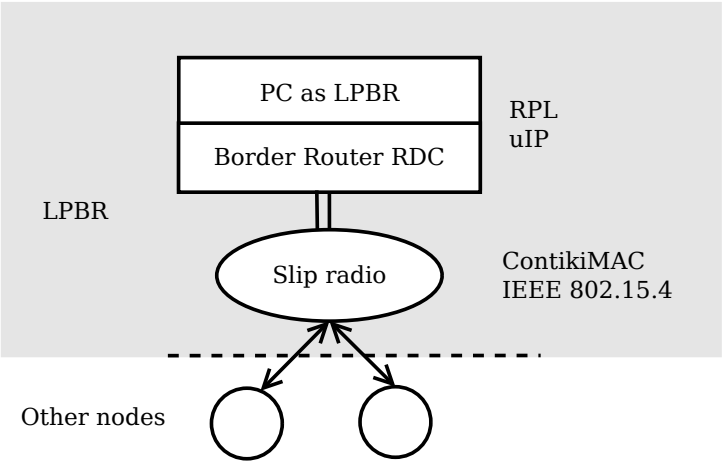


Figure 4.1: Low power border router

embedded PC over a serial line. The other side of the serial line does a similar job to bridge the embedded PC to the LPBR using the network interface. It constructs a SLIP tunnel between a virtual network interface (tun) and SLIP, the physical serial interface to encapsulate and pass the IP traffic to and from the other side of the serial line. The tun interface is used as any real network interface such as for routing and traffic forwarding [79, 80].

4.2 MCRP Implementation

MCRP is implemented in Contiki and uses ContikiMAC as the MAC protocol, RPL as the routing protocol. ContikiMAC is modified to allow multi channel where the channel selection processes take place on the upper layers and the channels are kept in the network neighbour table to ensure the correct channel.

The protocol implementation is separated into two types of nodes: i) the centralised LPBR where the bridging takes place between the border router on a PC to the nodes, and ii) the decentralised transmission nodes referred as other nodes. The implementations for both types are described below.

4.2.1 Low Power Border Router

As sensors have limited memory, most processing decisions at LPBR are transferred to a PC as it has more RAM and better processing capabilities. This enables MCRP

to have more thorough processes and to run in real time without draining the memory and battery on a sensor. LPBR is divided into two main parts as shown in Figure 4.1 where the PC is responsible as the application, transport, network and routing layers while a sensor (labelled slip radio) is set as the wireless interface to enable the PC to communicate with the other nodes via Contiki tunslip tool.

The LPBR acts as the tree root in RPL where it will initiate the creation of the RPL routing tree. LPBR is a special case as channel changes at LPBR is not as direct as other sensor nodes due to these two parts. However, it works similar ways to the other nodes.

LPBR main responsibility is to decide on the new channel selection. LPBR has no knowledge of all the channels condition at this point, thus, a channel is selected at random. LPBR keeps the results from the channel changes processes and based on it when selecting a new channel for the next node to ensure the new channel is at least two-hop away from another node using the same channel. This is done to ensure that the nearby nodes do not communicate of the same channel and risk interfering with each other.

Algorithm 1 Pseudo-code for two-hop colouring algorithm

Notations
R is a node that is a Route
N is a node Neighbour
RN is the Route's Neighbour node
currentCh is the node current listening channel
newCh is the new channel the node will change to
Pseudo-code
if *R currentCh* $\neq$ *newCh* **then**
    succeed one-hop
    check all *RN* channels
    if *RN* channel $\neq$ *newCh* **then**
        succeed two-hop
        confirm *newCh*
    end if
else
    generate a new *newCh*
    update the number of *newCh* generated for *R*
    use default channel 26 is all tries fail
end if

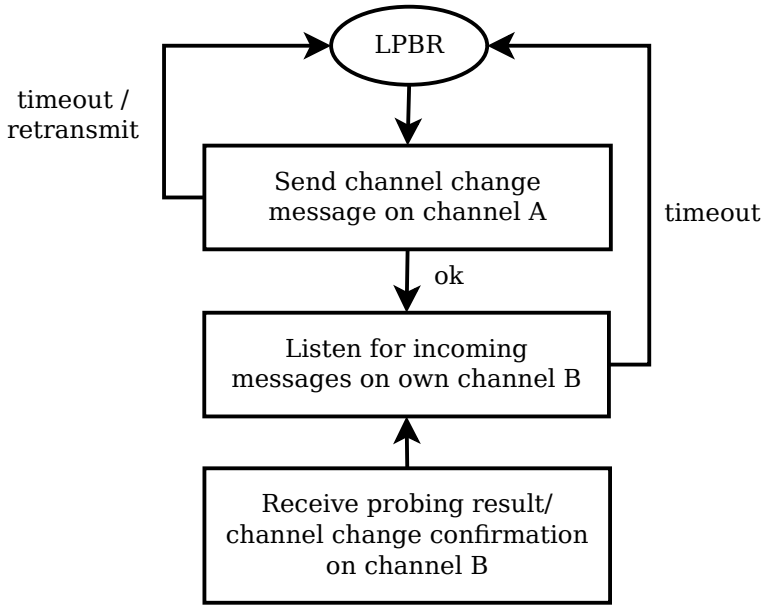


Figure 4.2: LPBR processes

The pseudo-code of the implemented two-hops colouring algorithm for new channel selection is shown in Algorithm 1. When the new channel is selected, LPBR will send the value to the intended node.

The new channel is stored in the buffer before the data is sent over SLIP to the radio-chip (slip-radio). As the slip radio is unable to access the neighbour table where the next hop node channel is stored, the channel value is passed through the buffer. LPBR keeps the updated value of all its neighbours channels in the neighbour channel. Slip radio that receives the packet buffer can access the channel value and kept the value is a simplified version of the neighbour table. This is done in order to ensure that the packet that is being queued or retransmitted is sent on the correct channel. The packets destination, which in this case, the next hop node is first check before the packet is transmitted each time. The MAC layer sets the channel accordingly before sending. ContikiMAC can access the simplified neighbour table as it is on the slip radio. The simplified neighbour table only keeps the information of the node neighbours and neighbours channels which are the critical information in order to transmit packets correctly. The other information that is related to the neighbours' conditions is monitored at the PC.

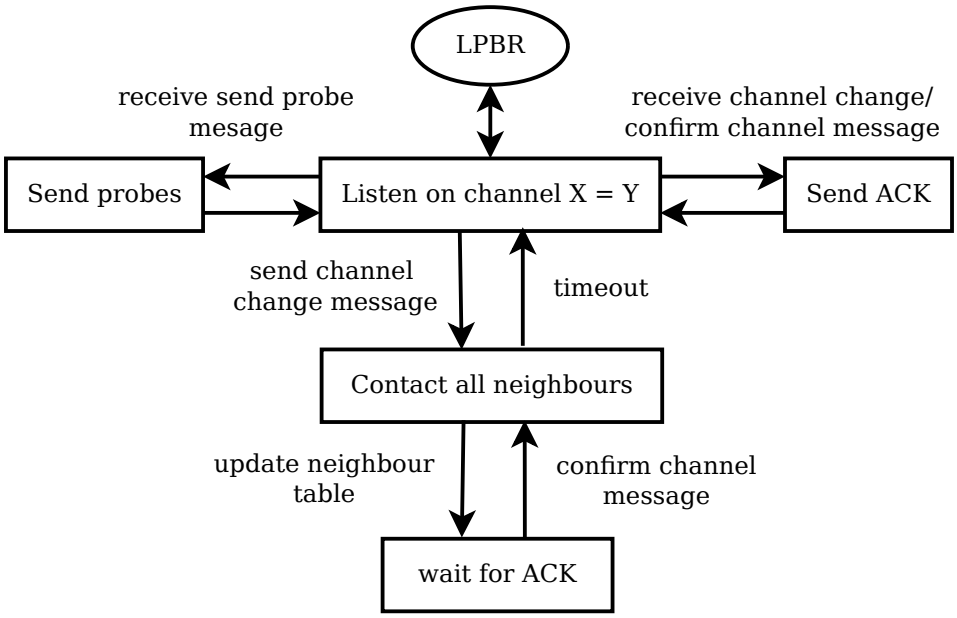


Figure 4.3: Nodes channel change processes

LPBR processes are shown in Figure 4.2. The slip-radio resets to its listening channel after the packet is transmitted. LPBR will wait and listen to any incoming packets. In the channel probing phase, LPBR does not take part in probing. However, LPBR is informed of the results of probing and kept a table of the probing results and channels to be able to use the information when deciding on a channel change based on the previous results of probing on the known channels.

4.2.2 Other Nodes

The new channel from LPBR that is received by the destination node is saved. Figure 4.3 shows the processes that the node takes in the channel change. When LPBR sends a *Channel Change* message to the destination node, the destination node will send a packet back to LPBR to acknowledge the channel change message. If LPBR does not receive the message, the channel change message is retransmitted. LPBR will then wait and listens for any incoming packets. At this point, channel changes processes will take place between the node and its neighbours. To clarify, *node* refers to the node that would like to change its listening channel and *neighbour node* is the neighbour of the node (including route node) that takes part in the channel change decision through probe message.

Unlike LPBR, other nodes have all the layers within the nodes themselves. This makes channel changes less complicated, however, the nodes are being limited by the number of RAM they have which resulted in probing values to be stored in the centralised LPBR. The nodes however, keep the probing results temporarily before the final decision of the channel is made.

The node sends the *Node New Channel* value to all of the node's neighbours. At this point, the new channel is not yet checked for its validity. However, all neighbours need to know the new channel as the node will change its listening channel to the new channel. Otherwise, packets cannot be received by the node since the listening channel is different than it was previously. The neighbours that receive the node's channel will update their *neighbour table* which is accessed from the application layer. Unlike LPBR, other nodes have all the layers within the nodes themselves. This makes channel changes less complicated as the nodes can access the information from any layer when required which in this case, accessing the neighbour table at the network layer from the application layer. In the neighbour table, a new entry is added to hold the channel value called *nbrCh*. As this is an important step in order to reduce the number of packet loss due to sending on the wrong channel, neighbours will send an acknowledgement of the new channel. Otherwise, it will be retransmitted.

The node will then send a *Start Probe* message to the neighbour that is a route node to start sending probing messages on the new channel. Not all neighbours are used as routes. The neighbours are chosen as route based on RPL OF which for this experiment is the ETX. The node will listens on the new channel and wait for the *Neighbour Probe* message. The route node starts to *Send Probe* messages every 3 seconds to allow retransmission or collision that could happen due to the busy channel. The maximum number of retransmission is configured to 3 attempts following the default value Contiki suggested. Collisions happen when the channel check keeps failing and new packets are constantly generated which could end up in a loop where no packets can be sent. The assumption that was made in this case is the channel will be cleared at some point which this loop will not happen. From

the experiments and simulations, this was proved true.

As only a small number of *Send Probe* messages are sent, the number of retransmissions and collisions that happen during the probing process are included in the channel decision process as it affects the channel reliability. As the retransmission and collision is a link layer process, the values are kept in a temporary *Retransmit Table* and is included to be sent in the next *Send Probe* message. This is because the value is only valid for that run. It gets reset each time a new packet is sent or received. The table is accessed at the application layer before the next *Send Probe* message is sent. The *Send Probe* message includes the current number of probe message and the number of tries (retransmissions and collisions) the previous packet had taken before it is successfully received. These values are used to decide if the channel is better than the current channel by giving a good probing result, meaning less retransmission.

The node keeps the value of all probing messages it receives. It sends the *Probe Result* message to LPBR. Unlike LPBR, the node has a limited RAM which resulted in past probing values to be stored in the centralised LPBR. The node however, keeps the probing results temporarily before the final decision of the channel is made. LPBR could use the information from the node's *Probe Result* to decide on a channel or blacklist bad channels.

The node then uses the values to decide whether the new channel is better than the previous channel by setting a threshold. The node then send *Confirm Channel* message to all neighbours that the node confirms to be listening on. The channel can be the new channel or the node can revert to the previous channel depending on the *Probe Result*. The neighbours will send an acknowledgement back to the node confirming the change. This is also important to ensure that all neighbours could communicate with the node on the correct channel. The neighbours will update their neighbour table of the node channel.

4.2.3 MAC Layer

As explained in Section 4.1.2, packets that have not been transmitted are queued in the buffer. The transmitting channel is set at the MAC layer as packets are not send

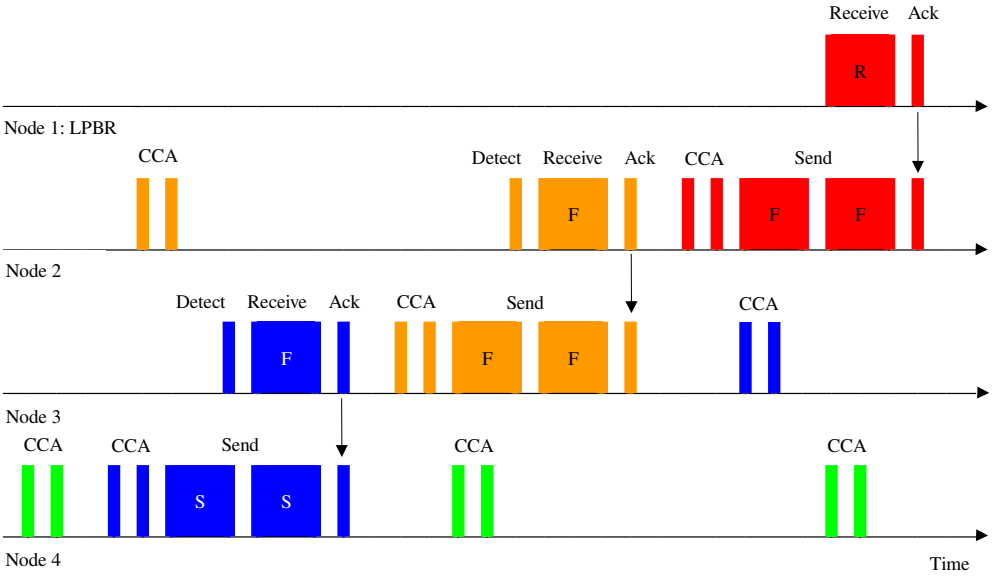


Figure 4.4: Multi channel ContikiMAC multi hop packet transmission

immediately if there are packets being queued. ContikiMAC is a single channel protocol. It is modified to support multi channel while complying to the same low power ContikiMAC principle. Each time the node goes to sleep, it will wake up on its listening channel waiting for incoming packets. If the node has a packet to send, it needs to change to the transmitting channel.

In order for the packet transmission or retransmission to be on the correct channel, the neighbour channel saved in the *neighbour table* at the network layer is accessed from the MAC layer and the channel is set to the transmitting channel. The node resets the channel to its listening channel after the transmission succeeded and goes back to sleep.

Figure 4.4 shows an example of a multi hop packet transmission in MCRP. The different colours represent different channels. Node 4's channel is represented by green, node 3 is blue, node 2 is yellow and node 1 is red. At each hop, the node changes to the next hop listening channel before forwarding the packet. In the example, node 4 is sending a packet to node 1, the LPBR through node 3 and node 2. Node 4 wakes up and checks for incoming packets on its channel. As it has a packet to be sent to node 1, it checks the next hop channel which is node 3 and changes the channel to node 3 listening channel. It checks if the channel is clear for transmission

and proceed to send the packet to node 3. Node 3 detects the packet when it wakes up and receives the packet. Node 3 sends the link layer acknowledgement to node 4 so that node 4 stops sending the packet. Node 3 forwards the packet to node 2 on node 2 channel and node 2 to node 1, the LPBR which is the destination node. All nodes reset their channel after the transmission and wake up on their own listening channel.

4.2.4 Network Layer

RPL is explained in Section 2.4. RPL control messages are tailored to accommodate MCRP proposal. Two main changes to RPL control messages are the DIS (which is sent by a new node to make it possible for a node to require DIO messages from a reachable neighbour) and DIO (the main source of routing control information) control messages. DIS and DIO control messages are usually sent using broadcast. However, DIS and DIO support unicast. MCRP sends RPL DIS control message in broadcast and DIO in both broadcast and unicast.

In MCRP, a new node that would like to join an existing tree needs to send the DIS control message to the reachable neighbours. However, as the reachable neighbours could be on different channel than it were initially during start up, the new node needs to send the DIS message on all channels available to be able to find the neighbours.

The neighbours that receive the DIS message will reply with a DIO message and a packet that tells the new node of its channel to communicate on. The new node updated the neighbour table and has successfully joined the tree.

If the neighbours do not receive the DIS from the new node before it is due to send the DIO message, the neighbours send a broadcast DIO on the default channel. The new node upon receiving the DIO will join the tree and updates the neighbour table. All neighbours send a DIO broadcast on the default channel and a DIO unicast for known neighbours on channels that the neighbours are listening on.

One of the main reasons for this is because broadcasting on all channels would require more energy and it would take a longer time before all reachable nodes receive the control message. This could delay changes that might happen in the

tree, i.e. changing of parent node. Secondly, all nodes by default will switch on to the same default channel as that is how the nodes are being set up.

Chapter 5

Results and Discussions

This chapter presents the evaluation of MCRP. The experiment set up for Cooja simulation and FlockLab [81] testbed are explained below. In Cooja simulation, interference is introduced. MCRP is evaluated using an end-to-end packet delivery performance metric. The results from the experiments are presented and discussed.

5.1 Experimental Setup

MCRP is evaluated in Cooja simulated environment and FlockLab testbed. In Cooja simulation, an interference model is used as simulation allows full control over the test environment and the experiments are repeatable. Although the interference model does not fully mimic the behaviour of real world interference, it enables MCRP performance to be tested in various conditions when the channel performance degraded. Unlike simulation, testbed provides the ability to validate MCRP performance in the real wireless channel environments. However, the network's behaviour are complicated to examine, thus, comparing the testbed result with simulation results give a better understanding of the performance.

5.1.1 Simulation

MCRP is evaluated in the Cooja simulated environment with emulation of TMote sky nodes that feature the CC2420 transceiver, a 802.15.4 radio. The nodes run on IPv6, using UDP with standard RPL and 6LoWPAN protocols. The network consists of 31 nodes which are used to run the simulation where one node is used as the border router node, 16 interference nodes, and 14 duty cycled nodes that

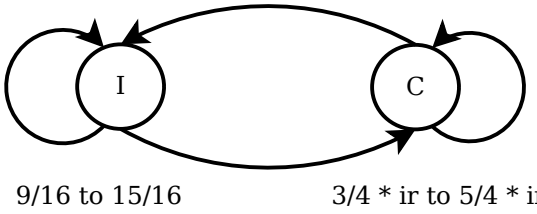


Figure 5.1: Low power border router

act as UDP clients to send packets to LPBR spanning over 20-30 metres between each node. RPL border router is used as LPBR in order to move most processing decisions on a PC as it has more RAM and better processing capabilities than a sensor. TelosB has limited RAM and ROM of 10K bytes and 48K bytes of flash memory [46]. By using a border router, this allows channel changing to be decided in real time without draining the memory and battery on a sensor. The border router also acts as the root of the tree.

A controlled interference node that generates semi-periodic bursty interference is simulated to resemble a simplified Wi-Fi or Bluetooth transmitter on several channels at random. The interference model proposed in [2] is used in the simulation. The interference has two states, a clear state and an interference state. In the interference state, the interference node generates packets for a time that is uniformly distributed between $9/16$ seconds and $15/16$ seconds. In the clear state the interferer produces no packets and stays in this state for between $3/4 * clear\_time$ and $5/4 * clear\_time$ where *clear_time* refers to the rate of interference. The model is illustrated in Figure 5.1. Multiple channels interference is used in the simulation to show the hypothesis that MCRP can help avoid interference. The scenario that is considered is where ContikiMAC with RPL system is subject to interference on its channel after set up has successfully completed so the RPL set up is allowed to complete before interference begins.

The protocol performance in loss over time in the presence of interference is observed. Two multiple channels interference scenarios are considered; (1) extreme and no interference rate on 8 channels each and (2) extreme, moderate, mild and no interference rate on 4 channels each. The interference channels are randomly cho-

sen from the available 16 channels and the same interference channels and rates are used throughout the experiments. However, channel 26 is kept clear from interference in order to ensure RPL set up is unaffected. In scenario 1, the interference rates are fixed to extreme and no interference to observe the effect it has on the channel changing decisions. In scenario 2, the interference rates are vary to observe how MCRP copes in deciding a channel when there is more interference than scenario 1 but with less interference intensity.

The simulation runs for a duration of 45-60 minutes to send 210-560 packets. When the nodes are switched on for the first time, all nodes are initialised to channel 26, the default channel for Contiki MAC layer. RPL is allowed five minutes to set up (which is ample time). RPL topology is formed in a minute. The simulation waits for another five minutes to allow trickle timer to double the interval length so that RPL control messages are not being sent frequently. The multichannel protocol is then runs for 25 minutes. In the 15 nodes simulation, the protocol takes 20-25 minutes to run the channel change set up. Another 5 minutes wait time is allowed if retransmissions happen. In a single channel simulation, all the nodes are changed to channel 22 after 5 minutes of RPL set up time. This allows RPL to have enough time to discover all nodes to form an optimised topology. The topology formation does not form completely if the interference node interferes from the beginning. The interference node starts sending packets to interfere after 3 minutes the system is switched on so that the interference channel is involve in the channel changes decision. It is proven that the protocol tries to avoid changing to the interference channel through time out and probing failures. After 30 minutes, the client nodes will send a normal packet periodically every 30-60 seconds to LPBR. This is done in order to avoid collision of the nodes sending at the same time.

5.1.2 Testbed

Similar to the simulation settings, the nodes run on IPv6, using UDP with standard RPL and 6LoWPAN protocols. The network consists of 26 nodes; 1 border router node and 25 duty cycled nodes as shown in Figure 5.2 where the grey lines represent the link qualities between nodes and yellow gradient is the noise. The indoor

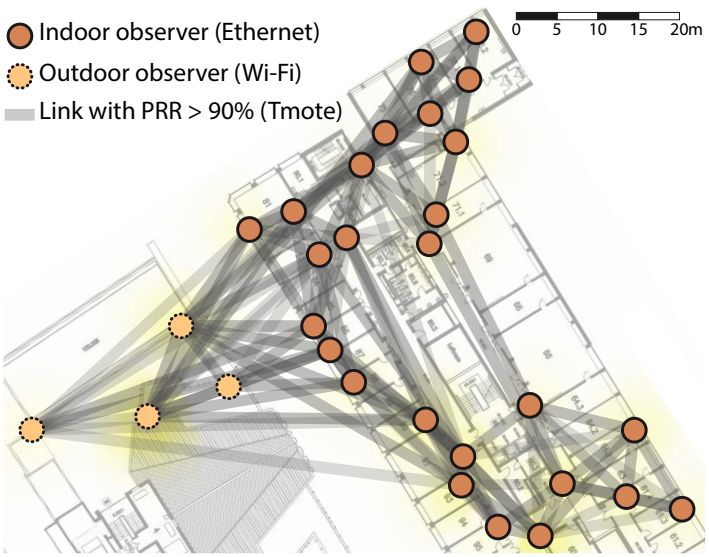


Figure 5.2: Layout of FlockLab deployment

nodes are distributed in offices, hallways and storerooms across one floor in an office building. Unlike in simulation, interference nodes are not added as the testbed has external interference from the surrounding (e.g. from the co-located Wi-Fi or limited connectivity) to see the effect on MCRP. The link qualities estimation information of the testbed is displayed on the FlockLab website. The link qualities and routing change throughout the day as the noise level on all channels and frequency bands are tested during testbed idle times.

As the channel condition in the testbed is beyond control, the experiment runs for a longer time period than the simulation. The testbed experiment is in the preliminary stage. The testbed is run for a duration of 4 hours to send up to 1250 packets depending on the availability of the nodes. The nodes are not switched on at the same time, thus RPL is allowed ten minutes to set up. MCRP is then run for 2 hours where the channel changes are done much slower than in the simulation as the channel condition is unknown, thus the number of retransmissions and collisions increases. The nodes wait for the normal incoming packet if the channel change processes finish early.

5.2 Evaluation

The performance of MCRP is compared against the standard ContikiMAC with RPL. MCRP is analysed using an end-to-end packet delivery performance metric. The transmission success rate is calculated from the sender to the receiver over multiple hops.

The simulations are repeated ten times. In all plots, the mean value of the ten simulations is plotted with error bars corresponding to one standard deviation in either deviation to give a measure of repeatability. The plots are of the proportion of received packets (from 0% to 100%) against time where the loss is measured over the previous time period. The x-value is shifted slightly left and right to prevent error bars overlapping.

5.2.1 Packet Loss Rates

The performance obtained in ContikiMAC with RPL (single channel) is compared with MCRP in terms of packet loss rate. As described previously, levels of interference used (referred to as *clear_time* in [2]) vary between 100% (no interference), 75% (mild), 50% (moderate) and 25% (extreme) where the percentage is the ratio of the time the channel is clear for transmission. All of the tests have a common format: the RPL procedure is allowed to set up without interference in order not to bias subsequent tests. Then the interferers begin to operate with a constant level (none, mild, moderate or extreme).

Figure 5.3 shows the results in simulation for ContikiMAC with RPL protocol. It can be seen that the level of packet loss varies considerably between experiments (the error bars are always large). It can also be seen that even for mild interference there is considerable loss and this gets worse as time proceeds. In the extreme interference case the loss always goes up until no packets are received. For mild interference the system evolves until it is losing around 20% of packets but this can increase.

The results from the single channel with interference is compared with the multichannel with the same interference rate of 75% (mild), 50% (moderate) and 25% (extreme). The test is done to evaluate MCRP behaviour in different interference

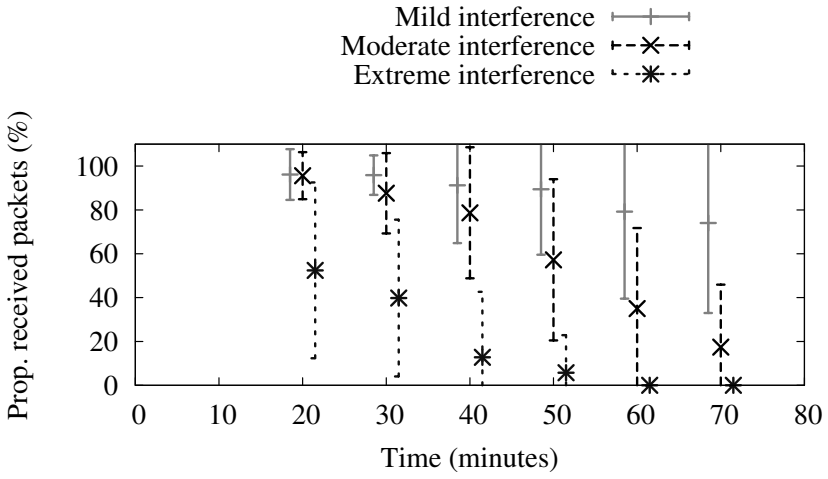
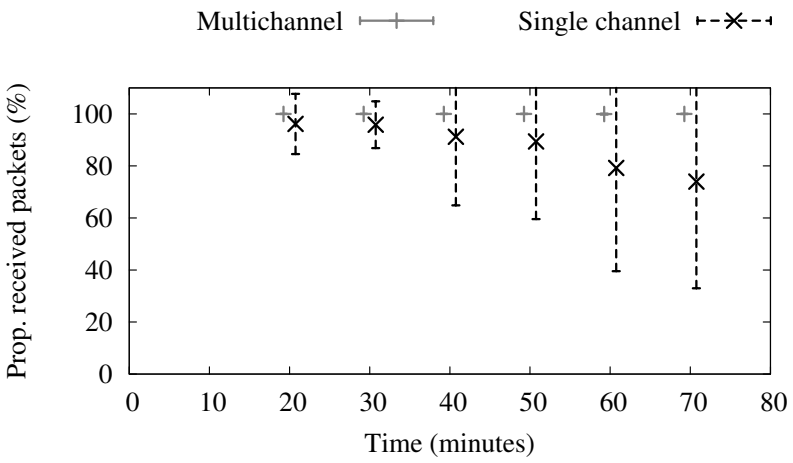


Figure 5.3: Level of packet loss for mild, moderate and extreme interference levels using single channel

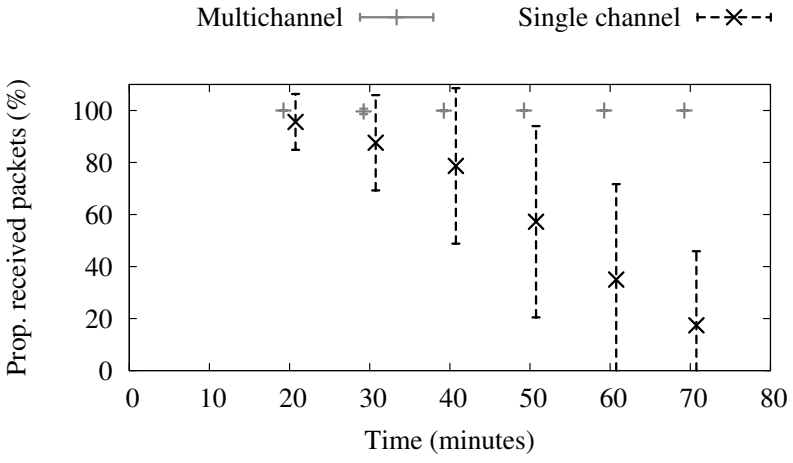
rate and to compare the result with a single channel case.

Figure 5.4 shows the averaged results from ten runs that were done. It can be observed that during high and moderate interference, if LPBR tries to send a channel change value that is the same channel as the interference, the request will either timed out or if it succeeds, the probing messages received are less than a threshold that allows for the node to change its listening channel to the new channel. This is as expected as MCRP checks the channel each time before deciding on the new channel to avoid interference channel. By doing this, it ensure that the node's listening channel is a good channel. This enable the use of all available channels without blacklisting any channel until it is sure that it is a bad channel through the probing process. The channel quality table is built at the LPBR that over time can be used to learn good and bad channels based on several probing processes. MCRP avoids the interference channel which as a result, resulted in less loss than in a single channel case.

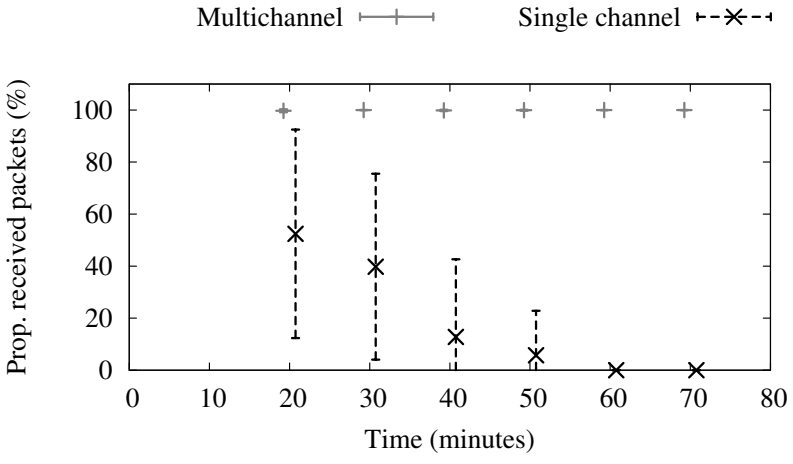
In the single channel, the node does not have enough time to recover from the interference to retransmit and drops all packets. Figure 5.4(c) shows that there are more packets drop over time and it stops receiving packets as it doesn't have enough buffer to store the incoming packet and the channel becomes congested. However, as the interference rate increases (less interference), the single channel performance improves as it has more time to recover.



(a) Mild Interference



(b) Moderate Interference



(c) Extreme Interference

Figure 5.4: Results of Multichannel RPL and a single channel RPL on different interference rate.

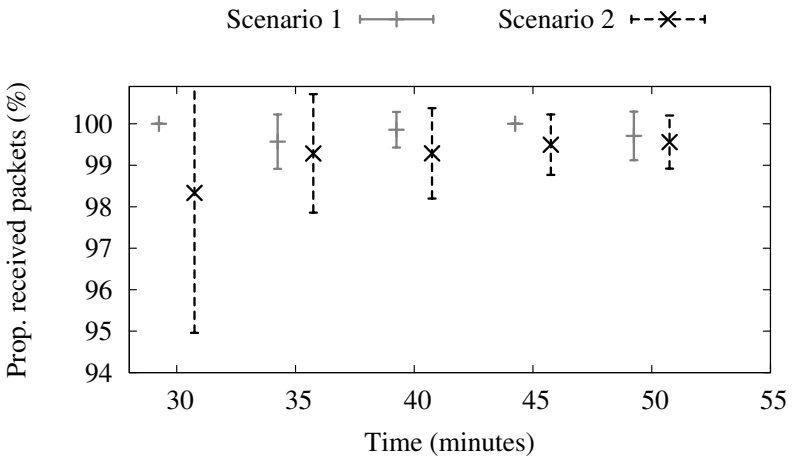


Figure 5.5: Level of packet loss for scenario 1 and scenario 2 using multi channel

In the mild interference case, all probing messages are received even though there is interference in that channel. This means that the channel can be used for transmission. As the interference rate is mild, all packets are received. This is also the case with a single channel. The interference does not affect the transmissions as the interference is not frequent enough. The node has enough time to recover from the interference through retransmissions. However, the interference would slightly effect the packet transmission over time. The channel change processes should run periodically to avoid this from happening.

To further evaluate MCRP capabilities to cope with interference from many sources, thus channels, two interference scenarios are considered. In scenario 1 half the channels (including the original channel) have no interference at all and half the channels have extreme interference. In scenario 2, four channels (including the original channel) have no interference, four have mild, four moderate and four extreme interference. Figure 5.5 shows multi channel results for these two scenarios. In scenario 1 the protocol performs extremely well, the packet loss is near zero and the protocol successfully detects channels with interference. Scenario 2 has similar results as in scenario 1. The protocol does well at reducing the effects of interference and could detect moderate and mild interference.

In the testbed case, the conditions for all channels are looked into to have a better idea of the variation in interference. The channels are tested on all channels

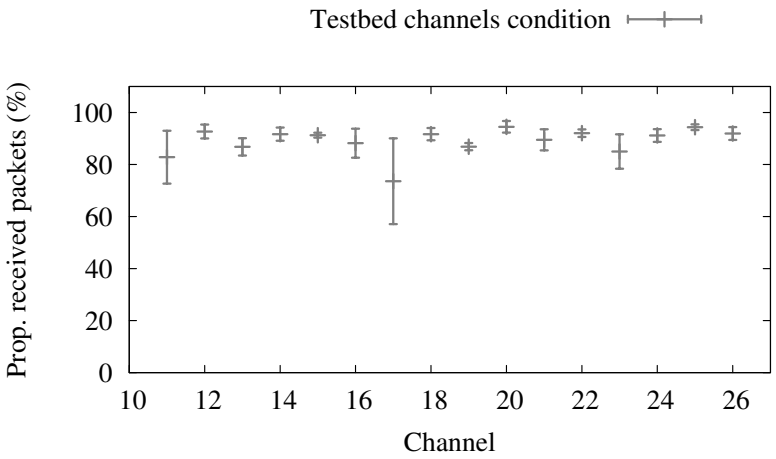


Figure 5.6: Level of packet loss on testbed for different channels

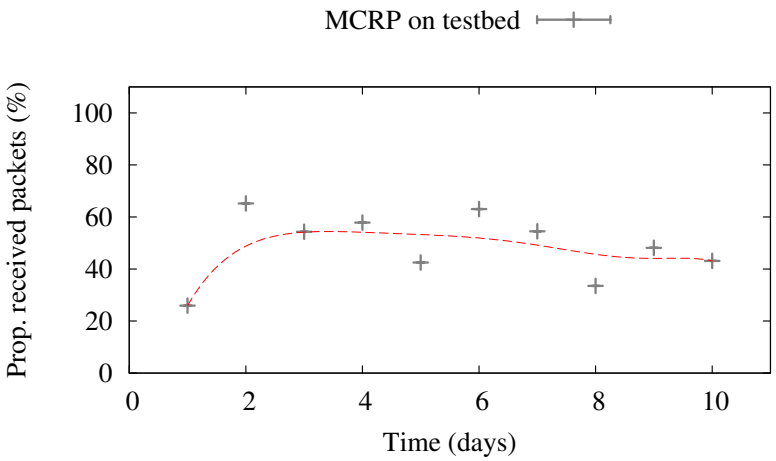


Figure 5.7: Level of packet loss on testbed using multi channel

for three different days. Figure 5.6 shows that the interference on the testbed varies for different channels on different days. However, it shows high receiving rate of 50%-90% in general.

Figure 5.7 shows the results from ten runs and the average that were done on different days. It can be observed that the results vary and multi channel give unstable results. On average, 50%-60% of packets are received at the LPBR. There are a few reasons for this low receiving packet values. As explained in Section 4.1.2, the incoming and outgoing packets are kept in the same buffer. When the channel is busy the outgoing and incoming packets will fill up the buffer which results in new packet to be dropped. The dropped packets might be the important packets that decide or confirm the new channel. This results in sending in incorrect channel.

The node will tries to send until it receives the link layer acknowledgement or time out which as a result, blocks the channel during its sending period by making the channel busy to be used by other node of the same channel. Another option to overcome this problem is by enabling RPL DIO on all channel so that the lost nodes can be recovered and updated with the correct channel each time the control message is sent. RPL DIO control message is sent to the neighbouring nodes according to the Trickle timer as explained in Section 2.4.

As the environment vary, it is not possible to replicate the same experiment as the topology might have changed depending on the RPL ETX value at each run. The nodes are possibly to take different routes to send the packet than in previous experiment.

In MiCMAC [8], it is stated that MiCMAC has a transmission success rate of 99% when using four channels. However, when more than four channels are used (8 or 16 channels), MiCMAC performance degrades to approximately 88% (16 channels) due to interference channels. The interference model that MiCMAC uses is different than in this experiment. They compare the result with Chrysso where Chrysso has a transmission success rate of approximately 88% for 4 and 8 channels and suffers greatly in the case of 16 channels with 60% success rate. MCRP on the other hand, shows greatly reduced loss rate with any number of channels at approximately 99%.

5.2.2 Setup Overhead

Obviously the system of changing channels and probing to see if a channel is free of interference introduces a certain amount of overhead into the protocol. This takes the form of (a) extra messages passed and (b) extra time taken to set up. Default RPL on ContikiMAC for the topology considered in these experiments completed its set up using 276 packets. MCRP, the multi-channel protocol completed its set up in 716 packets, that is an overhead of 440 packets on top of RPL. This overhead comes from the channel changing messages to nodes and neighbours, probing messages, channel confirmation messages and acknowledgement packets which are required to ensure a thorough channel change decision. However, it is worth mentioning that

this is a one-off cost. This represents (in this experimental set up) approximately one hour of extra packets in the situation of a deployment that is meant to work for weeks or months. In terms of set up time, the protocol begins to change channels only when the RPL set up process is complete (or at least stabilises). The set up time is 1154 seconds beyond the RPL set up time of 286 seconds. However, it should be noted that, in fact, the system remains fully functional and capable of sending packets during the set up so this set up overhead does not matter to data transmission. Therefore it can be concluded that data sending costs (extra packets) of set up are negligible in the context of a deployment that will last more than a day. The extra set up time is also negligible within this context and furthermore does not degrade performance of the network during this set up phase.

Chapter 6

Energy Efficient WSNs

There have been many works done to study and estimate the nodes energy consumption in real time in order to prolong the network lifetime. There are three main ways that have been exploited for an energy-efficient WSN which are through MAC protocols and routing protocols. However, most solutions use other metrics to represent the energy consumption such as the radio duty cycle and end to end delay as it is more complex to compute the energy level from the battery-powered sensors.

6.1 Existing Energy Efficient Solutions

Energy consumption estimation is often used for comparison between different nodes thus the voltage is not required to be computed. It is possible to measure the battery level for battery-powered sensors, however, it cannot be directly translated for energy estimation because of the non-linearity of batteries.

6.1.1 Energy-based RPL

RPL is a routing protocol that builds the topology based on the routing metrics and constraints for path calculation that are defined separately from the topology. This separation allows new metrics and constraints to be defined to fulfil the specific application and network optimisation criteria. A routing metric is used to evaluate the path cost. The routing metrics can be categorised into link and node metrics [66]. In node metrics, it can be the node state which provides information about the node characteristics, energy such as selecting nodes with higher residual energy or hop count. In link metrics, it includes the link throughput, latency or link reliability

such as ETX. RPL lists the metrics that could be used. However, the implementation is left to the application.

[82] designed an OF for RPL that uses the node remaining energy as the metric during the parent selection of the topology. It aims to select nodes with higher remaining power level as the path for transmissions. The implementation uses a battery theoretical model [83] to estimate the node's battery lifetime at runtime. The OF concentrates on the node battery level estimation, path cost and node rank computation in selecting a parent. The node that advertises the maximum greatest path cost is selected as the parent. The maximum path cost from the node to the sink is computed as the minimum node energy level.

6.1.2 Real Time Energy Estimation

Powertrace system [98] is able to profile the power behaviour at the network layer for WSNs. It computes the energy consumption estimation of sensors at run time that depends on the software based on-line energy estimation mechanism [84] and able to show per-activity power cost such as the different states for wake ups, transmissions and receptions of a node. The software based on-line energy estimation mechanism is used to estimate the node's current energy consumption in real time. The on-line energy estimation is implemented in Contiki OS. The energy estimation module uses time measurements that can be directly obtained from the microprocessor on-chip timer. When the component is switched on to produce a time stamp. The time difference from when the component was on and when it later is switched off is computed. The current draw of the component is used to compute the total energy consumption estimation.

[22] developed a generic method to predict a node's energy consumption by capturing the interference patterns. The interference characteristics of a specific deployment site are captured to enable estimation of the node energy consumption when nodes are deployed at the same location in the future.

These energy consumption estimation solutions can be used to improve the network by using the information to reconstruct the topology.

6.2 Energy Efficient MCRP

An ongoing work in MCRP is in deciding a routing protocol that considers the energy consumption of the nodes in the network. MCRP is a centralised channel switching process at the LPBR. As LPBR is fully powered and has unlimited memory, LPBR has the ability to compute all the nodes energy level. The topology tree is routed at LPBR, thus LPBR is in charge to reconfigure the tree depending on the energy level that it computed. By using the energy estimation model, the nodes can send the measurements to LPBR. This moves the computation load from the nodes to LPBR. Based on this information, the LPBR can compute the energy level and it will know all nodes energy level.

$$\sum_{l \in links} = (c_l + c'_l) d_l \tag{6.1}$$

Consider the energy as the sum of all links used by a node; where it depends on the cost per message of the link, c_l and the total descendent that can be reached by the link, d_l where it is assumed that the number of sent and received packets are the same for each descendent. However, the cost of the links upwards (towards LPBR) is different from the cost of the links downwards as upwards and downwards links use different channels.

Based on the energy level of each node and cost of the link (link is weighted based on the transmissions rate on the link channel) LPBR can sends a reconfiguration message so that the nodes will route using the nodes and paths that are more energy efficient.

Chapter 7

Future Work

7.1 Conclusions

WSNs are widely used in many crucial applications such as in remote environmental monitoring and target tracking as sensors can easily be deployed in difficult locations. However, WSNs suffer from sensors limited hardware and energy capabilities, and the unreliable network environment which impact the sensors performance, thus the efficiency of the network.

In this work, MCRP is presented. MCRP is a decentralised cross-layer protocol with a centralised controller. The protocol mitigates the effect of interference by avoiding the affected channels through channel switching processes. It allows better spectrum usage by moving nearby nodes to listen on different channel using two-hop colouring algorithm. MCRP provides feedback when a channel is subject to interference using the probing phase. The results from the simulation showed that MCRP avoids channels with interference hence greatly reduced loss rate with negligible overhead. By reducing packet loss (hence retransmissions) and increasing the efficiency of spectrum usage, the multichannel system will be more energy efficient than single channel ContikiMAC with RPL over the lifetime of the system's deployment.

7.2 Future Works

Future work is ongoing to develop the protocol. Deployment is underway on the Flocklab testbed as adjustments are required to enable MCRP to provide similar re-

sult as the simulation. This is due to unseen problems that do not occur in the simulation environment such as frequent nodes disconnection, buffer overload, change of paths and link conditions that vary throughout the day. The protocol is also planned to be tested on real hardware locally where the environment condition can be controlled where a better interference model can be used to closely replicate the real world environment in the case of extreme interference such as at a busy train station area.

The protocol will be tested against competing multichannel protocols to prove that channel selection at run time have better result, efficiency and reception rate than blind channel hopping.

The protocol will be further developed to update the LPBR on ongoing packet loss so that the network can continually respond in changes in congestion. This will also enable certain channels characteristics and patterns to be learned over time for the specific location for better channel selection processes.

The protocol is currently looking into the network energy consumption. MCRP will consider the nodes energy level and the paths reliabilities in order to prolong the network lifetime without compromising the network efficiency.

Bibliography

[1] IEEE. IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, April 2012.

[2] Carlo Alberto Boano, Thiemo Voigt, Nicolas Tsiftes, Luca Mottola, Kay Römer, and Marco Antonio Zúñiga. Making sensornet MAC protocols robust against interference. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 272–288, 2010.

[3] M. Petrova, Lili Wu, P. Mahonen, and J. Riihijarvi. Interference measurements on performance degradation between colocated IEEE 802.11g/n and IEEE 802.15.4 networks. In *Networking, 2007. ICN '07. Sixth International Conference on*, pages 93–93, April 2007.

[4] IEEE. IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[5] Yafeng Wu, J.A. Stankovic, Tian He, and Shan Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, April 2008.

[6] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the*

6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, pages 116–123, 2009.

[7] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso - a multi-channel approach to mitigate external interference. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 449–457, June 2011.

[8] B. Al Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multi-channel. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 2–9, May 2014.

[9] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462, Nov 2004.

[10] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with COOJA. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 641–648, Nov 2006.

[11] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, August 2008.

[12] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhury. A survey on wireless multimedia sensor networks. *Comput. Netw.*, 51(4):921–960, March 2007.

[13] JeongGil Ko, Joakim Eriksson, Nicolas Tsiftes, Stephen Dawson-Haggerty, Jean-Philippe Vasseur, Mathilde Durvy, Andreas Terzis, Adam Dunkels, and David Culler. Beyond Interoperability: Pushing the Performance of Sensor Network IP Stacks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 1–11, New York, NY, USA, 2011. ACM.

[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Comput. Netw.*, 38(4):393–422, March 2002.

[15] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. *SIGCOMM Comput. Commun. Rev.*, 31(2 supplement):20–41, April 2001.

[16] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, March 2006.

[17] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. Monitoring behavior in home using a smart fall sensor and position sensors. In *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*, pages 607–610, 2000.

[18] Sibbald Barbara. Use computerized systems to cut adverse drug events: report. *CMAJ*, 164(13):1878, June 2001.

[19] E.M. Petriu, Nicolas D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *Instrumentation Measurement Magazine, IEEE*, 3(4):31–35, Dec 2000.

[20] Gyula Simon, Miklós Maróti, Ákos Lédeczi, György Balogh, Branislav Kusy, András Nádas, Gábor Pap, János Sallai, and Ken Frampton. Sensor network-based countersniper system. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 1–12, New York, NY, USA, 2004. ACM.

[21] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: Experiences from a

semiconductor plant and the north sea. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 64–75, New York, NY, USA, 2005. ACM.

[22] Alex King, James Brown, John Vidler, and Utz Roedig. Estimating node lifetime in interference environments. In *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*, pages 796–803, Oct 2015.

[23] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11):976–978, Nov 2005.

[24] Yi-hua Zhu, Wan-deng Wu, Jian Pan, and Yi-ping Tang. An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks. *Comput. Commun.*, 33(5):639–647, March 2010.

[25] Jie Wu, Ming Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *Parallel Processing, 2001. International Conference on*, pages 346–354, Sept 2001.

[26] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, Jan 2000.

[27] Li Li and J.Y. Halpern. Minimum-energy mobile wireless networks revisited. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 1, pages 278–283 vol.1, Jun 2001.

[28] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 70–84, New York, NY, USA, 2001. ACM.

[29] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. MAC essentials for wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 12(2):222–248, Second 2010.

[30] Lei Tang, Yanjun Sun, O. Gurewitz, and D.B. Johnson. PW-MAC: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1305–1313, April 2011.

[31] Youngmin Kim, Hyojeong Shin, and Hojung Cha. Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 53–63, April 2008.

[32] N.A. Pantazis, S.A. Nikolidakis, and D.D. Vergados. Energy-efficient routing protocols in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(2):551–591, Second 2013.

[33] Omprakash Gnawali. The minimum rank with hysteresis objective function, RFC 6719. https://tools.ietf.org/html/rfc6719, 2012.

[34] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, June 2005.

[35] W.K.G. Seah, Zhi Ang Eu, and H. Tan. Wireless sensor networks powered by ambient energy harvesting (wsn-heap) - survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, pages 1–5, May 2009.

[36] Z.G. Wan, Y.K. Tan, and C. Yuen. Review on energy harvesting and energy management for sustainable wireless sensor networks. In *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, pages 362–367, Sept 2011.

[37] James M Gilbert and Farooq Balouchi. Comparison of energy harvesting systems for wireless sensor networks. *international journal of automation and computing*, 5(4):334–347, 2008.

[38] Luigi Alfredo Grieco Thomas Watteyne, Maria Rita Palattella. Using IEEE802.15.4e time-slotted channel hopping (TSCH) in an internet of things (IoT): Problem statement. https://tools.ietf.org/html/rfc7554, May 2015.

[39] Ozlem Durmaz Incel, Lodewijk van Hoesel, Pierre Jansen, and Paul Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Netw.*, 9(1):73–94, January 2011.

[40] L.F.W. van Hoesel and P.J.M. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *1st International Workshop on Networked Sensing Systems, INSS 2004*, pages 205–208, Tokyo, Japan, 2004. Society of Instrument and Control Engineers (SICE).

[41] Adam Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report T2011:13. ISSN 1100-3154 http://dunkels.com/adam/dunkels11contikimac.pdf, 2011.

[42] Joseph Polastre, 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*, pages 95–107. ACM, 2004.

[43] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.

[44] David Moss and Philip Levis. BoX-MACs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, pages 116–119, 2008.

[45] Amre El-Hoiydi and Jean-Dominique Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31. Springer, 2004.

[46] Crossbow Technology. TelosB - TelosB mote platform. Document Part Number: 6020-0094-01 Rev B.

[47] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.

[48] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[49] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM, 2000.

[50] David Braginsky and Deborah Estrin. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM, 2002.

[51] Curt Schurgers and Mani B Srivastava. Energy efficient routing in wireless sensor networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, volume 1, pages 357–361. IEEE, 2001.

[52] Maurice Chu, Horst Haussecker, and Feng Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–313, 2002.

[53] Arati Manjeshwar and D.P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, April 2001.

[54] Narayanan Sadagopan, Bhaskar Krishnamachari, and Ahmed Helmy. The ACQUIRE mechanism for efficient querying in sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 149–155. IEEE, 2003.

[55] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[56] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16–27, Oct 2000.

[57] Jae-Hwan Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(4):609–619, Aug 2004.

[58] A. Manjeshwar and D.P. Agrawal. Apteen: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless. In *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM*, pages 8 pp–, April 2002.

[59] S. Lindsey and C.S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–1125–3–1130 vol.3, 2002.

[60] S. Lindsey, C. Raghavendra, and Krishna Sivalingam. Data gathering in sensor networks using the energy*delay metric. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2001–2008, April 2001.

[61] O. Younis and Sonia Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4):366–379, Oct 2004.

[62] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, 2009.

[63] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis S. Kim, and A. Woo. TEP 123: The Collection Tree Protocol, Aug 2006.

[64] T Winter, P Thubert, T Clausen, J Hui, R Kelsey, P Levis, K Pister, R Struik, and J Vasseur. RPL: IPv6 routing protocol for low power and lossy networks, RFC 6550. https://tools.ietf.org/html/rfc6550, 2012.

[65] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (RPL), RFC 6552. https://tools.ietf.org/html/rfc6552, 2012.

[66] J Vasseur, M Kim, K Pister, N Dejean, and D Barthel. Routing metrics used for path calculation in low power and lossy networks. https://tools.ietf.org/html/rfc6551, 2012.

[67] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Osterlind, Joel Hglund, and Adam Dunkels. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 479–480, New York, NY, USA, 2010.

[68] Tsvetko Tsvetkov. RPL: IPv6 routing protocol for low power and lossy networks. *Sensor Nodes–Operation, Network and Application (SN)*, 59:2, 2011.

[69] Philip Levis, T Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. RFC 6206: The trickle algorithm. https://tools.ietf.org/html/rfc6206, 2011.

[70] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[71] J Hui and P Thubert. Compression format for IPv6 datagrams over IEEE 802.15.4-based networks, RFC 6282. https://tools.ietf.org/html/rfc6282, 2011.

[72] Adam Dunkels. Full tcp/ip for 8-bit architectures. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 85–98, New York, NY, USA, 2003. ACM.

[73] Contiki. Contiki 2.6. http://contiki.sourceforge.net/docs/2.6/, Jul 2012.

[74] Contiki. Contiki 2.6 The uIP TCP/IP stack. http://contiki.sourceforge.net/docs/2.6/a01793.html, Jul 2012.

[75] Adam Dunkels. Rime - a lightweight layered communication stack for sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session, Delft, The Netherlands*, January 2007.

[76] Adam Dunkels. Contiki Crash Course, October 2008.

[77] Adam Dunkels, Fredrik Österlind, and Zhitao He. An adaptive communication architecture for wireless sensor networks. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 335–349, New York, NY, USA, 2007. ACM.

[78] J Romkey. A nonstandard for transmission of IP datagrams over serial lines: SLIP, RFC 1055. https://tools.ietf.org/html/rfc1055, 1988.

[79] FIT IoT-LAB. Building Contiki's tunslip6. https://www.iot-lab.info/tutorials/build-tunslip6/.

[80] David Carels, Niels Derdaele, EliDe Poorter, Wim Vandenberghe, Ingrid Moerman, and Piet Demeester. Support of multiple sinks via a virtual root for

the rpl routing protocol. *EURASIP Journal on Wireless Communications and Networking*, 2014(1), 2014.

[81] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 153–166, New York, NY, USA, 2013. ACM.

[82] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio, and Olivier Festor. Energy-based metric for the routing protocol in low-power and lossy network. In *SENSORNETS*, pages 145–148, 2013.

[83] Emmanuel Nataf and Olivier Festor. Accurate online estimation of battery lifetime for wireless sensors network. In *Proceedings of the 2nd International Conference on Sensor Networks*, pages 59–64, 2013.

[84] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 28–32. ACM, 2007.

[85] Thang Vu Chien, Hung Nguyen Chan, and Thanh Nguyen Huu. A comparative study on operating system for wireless sensor networks. In *2011 International Conference on Advanced Computer Science and Information System (ICACSIS)*, pages 73–78, December 2011.

[86] Lanny Sitanayah, Cormac J. Sreenan, and Szymon Fedor. A cooja-based tool for maintaining sensor network coverage requirements in a building. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 70:1–70:2, 2013.

[87] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, 2013.

[88] A. Sivanantha, B. Hamdaoui, M. Guizani, Xiuzhen Cheng, and T. Znati. EM-MAC: An energy-aware multi-channel MAC protocol for multi-hop wireless networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 1159–1164, Aug 2012.

[89] Asaduzzaman and Hyung Yun Kong. Energy efficient cooperative LEACH protocol for wireless sensor networks. *Communications and Networks, Journal of*, 12(4):358–365, Aug 2010.

[90] Joris Borms, Kris Steenhaut, and Bart Lemmens. Low-overhead dynamic multi-channel MAC for wireless sensor networks. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 81–96, 2010.

[91] A.A. Aziz, Y.A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 15(1):121–144, First 2013.

[92] An-Feng Liu, Peng-Hui Zhang, and Zhi-Gang Chen. Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks. *Journal of Parallel and Distributed Computing*, 71(10):1327 – 1355, 2011.

[93] An-Feng Liu, Peng-Hui Zhang, and Zhi-Gang Chen. Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks. *Journal of Parallel and Distributed Computing*, 71(10):1327 – 1355, 2011.

[94] Hongbo Jiang, Shudong Jin, and Chonggang Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064–1071, June 2011.

[95] Jalel Ben-Othman and Bashir Yahya. Energy efficient and QoS based routing protocol for wireless sensor networks. *J. Parallel Distrib. Comput.*, 70(8):849–857, August 2010.

[96] I.A. Essa. Ubiquitous sensing for smart and aware environments. *Personal Communications, IEEE*, 7(5):47–49, Oct 2000.

[97] Shio Kumar Singh, MP Singh, DK Singh, et al. Routing protocols in wireless sensor networks–a survey. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, 1:63–83, 2010.

[98] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. 2011.