UNIVERSITY OF ZULULAND

# Topology Control for Wireless Mesh Networks and its Effect on Network Performance

by

Pragasen Mudali

A thesis submitted in fulfillment of the
degree of Doctor of Philosophy
in the

Faculty of Science and Agriculture
Department of Computer Science

Promoter: Prof. M. O. Adigun
Co-Promoter: Dr. N. Ntlatlapa

November 2016

# *Abstract*

Infrastructure Wireless Mesh Networks (I-WMNs) are increasingly used to provide network connectivity and Internet access to previously under-served areas in the developing world. It is common for some of these deployments to be battery-powered due to a lack of electrical infrastructure in the targeted areas. Thus, the energy-efficiency of these networks gains additional importance.

Topology Control (TC) has been previously reported to improve the energy-efficiency and network performance of wireless ad-hoc networks, including I-WMNs. However, simulation-based studies have been relied upon to reach these conclusions and the study of TC prototypes applicable to I-WMNs has largely been limited to design issues. Thus, the study of the efficacy of TC prototypes as a mechanism for improving energy-efficiency and network performance remains an open issue.

The thesis addresses this knowledge gap by studying the dynamic, run-time behaviours and the network topologies created by two standards-compatible TC prototypes. This study provides unique insight into how the prototypes consume computational resources, maintain network connectivity, produce cumulative transceiver power savings and affect the workings of the routing protocol being employed.

This study also documents the topology instability caused by transceiver power oscillations produced by the PlainTC prototype. A context-based solution to reduce transceiver power oscillations and the subsequent topology instability is proposed. This solution applies the Principal Component Analysis statistical method to historical network data in order to derive the weights associated with each of the identified context variables. A threshold value is defined that only permits a node to adjust its transceiver power output if the observed change in a node's context exceeds the threshold. The threshold mechanism is incorporated into the PlainTC+ prototype and is shown to reduce topology instability whilst improving network performance when compared to PlainTC.

The results obtained in this study suggest that I-WMN topologies formed by TC are able to closely match the performance of networks that do not employ TC. However, this study shows that TC negatively affects the energy efficiency of the network despite achieving cumulative transceiver power savings.

# Declaration of Authorship

I, PRAGASEN MUDALI, declare that this thesis titled, "Topology Control for Wireless Mesh Networks and its Effect on Network Performance" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed: ______________________________

Date: ______________________________

*Dedicated to my Family*

# *Acknowledgements*

I would like to acknowledge the various people and institutions that have made this study possible. I am grateful for the assistance provided. My supervisors, Prof. Matthew Adigun and Dr. Ntsibane Ntlatlapa must be acknowledged for supporting my passion for this avenue of research. Prof. Adigun's light touch and thought-provoking feedback helped to guide this study. His efforts to secure the study leave required to complete this Thesis are greatly appreciated. Many thanks must also go to Dr. Ntlatlapa for supporting this study under the auspices of a studentship with the Meraka Institute at the Council for Scientific and Industrial Research (CSIR) in Pretoria.

This study was also supported by the Centre of Excellence for Mobile e-Services hosted within the Department of Computer Science at the University of Zululand. I am thankful to my colleagues in the Wireless Mesh Networks cluster within the Centre for their contributions to this study. Special thanks must also go to Thulani Nyandeni for his contribution to the conceptualisation of this study and to Edgar Jembere for his statistical insights and recommendation of the *R* statistical analysis tool. Thanks must also go to Lawrence Mboweni from the Meraka Institute for exposing me to the practical aspects of Wireless Mesh Networks. The insights gained were invaluable and contributed to the feasibility of this study.

I would like to acknowledge my family for encouraging me to pursue a Ph.D and for supporting me through this period. Thank you.

God bless you all.

# *List of Publications*[1]

**Journal Article**

Work from Chapter 7 and Section 6.2 appears in:
**P. Mudali**, M.O. Adigun. Context-based Topology Control for Wireless Mesh Networks, *Mobile Information Systems*, vol. 2016, Article ID 9696348, 16 pages, 2016.

**Conference Proceedings**

Work from Section 4.3 and Section 7.3 appears in:
**P. Mudali**, M.O. Adigun. Design Considerations of a Topology Control Prototype for Wireless Mesh Networks, *IEEE International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Durban, 28-29 November 2016.

Work from Section 4.4 and Section 6.3.7 appears in:
**P. Mudali**, M.B. Mutanga, M.O. Adigun, N. Ntlatlapa. Evaluating the Effect of a Topology Control Scheme on Application Layer Traffic Scenarios in Infrastructure Wireless Mesh Networks, *Southern African Telecommunications and Network Applications Conference (SATNAC)*, George, 5-8 September 2012.

Work from Section 3.2 appears in:
**P. Mudali**, M.B. Mutanga, M.O. Adigun, N. Ntlatlapa. Evaluating Transceiver Power Savings Produced by Connectivity Strategies for Infrastructure Wireless Mesh Networks, *International Conference on Wireless and Mobile Communications (ICWMC)*, pp. 215-220, 2011.

Work from Section 3.4 appears in:
**P. Mudali**, T.C. Nyandeni, N. Ntlatlapa, M.O. Adigun. A Test-Bed Evaluation of Connectivity Strategies for Infrastructure Wireless Mesh Networks, *Southern African Telecommunications and Network Applications Conference (SATNAC)*, Western Cape, 5-8 September 2010.

Work from Sections 3.3, 4.4 and 4.6 appears in:
**P. Mudali**, T.C. Nyandeni, N. Ntlatlapa, M.O. Adigun. Design and Implementation of a Topology Control Scheme for Wireless Mesh Networks, *IEEE Africon*, Nairobi, 23-25 September 2009.

[1] as at November 2016

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **CDS** | **C**onnected **D**ominating **S**et |
| **CNN** | **C**ritical **N**umber of **N**eighbours |
| **COTS** | **C**ommercial **O**ff **T**he **S**helf |
| **CTR** | **C**ritical **T**ransmission **R**ange |
| **GG** | **G**abriel **G**raph |
| **IEEE** | **I**nstitute of **E**lectrical and **E**lectronic **E**ngineers |
| **I-WMN** | **I**nfrastructure **W**ireless **M**esh **N**etwork |
| **LTE** | **L**ong **T**erm **E**volution |
| **MAC** | **M**edium **A**ccess **C**ontrol |
| **MANET** | **M**obile **A**d-hoc **N**etwork |
| **MAP** | **M**esh **A**ccess **P**oint |
| **MP** | **M**esh **P**oint |
| **MPP** | **M**esh **P**ortal **P**oint |
| **MST** | **M**inimum **S**panning **T**ree |
| **NVRAM** | **N**on-**V**olatile **R**andom **A**ccess **M**emory |
| **OLSR** | **O**ptmised **L**ink **S**tate **R**outing |
| **PDR** | **P**acket **D**elivery **R**atio |
| **RNG** | **R**elative **N**eighbourhood **G**raph |
| **RSSI** | **R**eceived **S**ignal **S**trength **I**ndicator |
| **SDN** | **S**oftware **D**efined **N**etworking |
| **TC** | **T**opology **C**ontrol |
| **TCP** | **T**ransmission **C**ontrol **P**rotocol |
| **UDP** | **U**ser **D**atagram **P**rotocol |
| **VANET** | **V**ehicular **A**d-hoc **N**etwork |

| | |
|---|---|
| **WAHN** | **W**ireless **A**d-hoc **N**etwork |
| **WiFi** | **Wi**reless **Fi**delity |
| **WiMAX** | **W**orldwide **I**nteroperability for **M**icrowave **A**ccess |
| **WMN** | **W**ireless **M**esh **N**etwork |
| **WSN** | **W**ireless **S**ensor **N**etwork |

# Part I

# Introduction and Background

# Chapter 1

# Introduction

Wireless Mesh Networks (WMNs) are being used as an alternative networking infrastructure to provide Internet connectivity in areas where it is currently economically unfeasible to deploy fixed-line or optical networks. WMN deployments are increasingly prevalent in rural, developing world scenarios due to the lower upfront costs and ease of deployment (Akyildiz and Wang, 2005; Bruno et al., 2005).

Rural WMN deployments are found to produce socio-economic benefits and lessen the urban-rural divide. Reports of educational, economic and social benefits can be found in the literature (Raman, 2004; Ishmael et al., 2008; Coates and Delwiche, 2009; Johnson, 2013). These deployments are often found in power-scarce areas and are thus battery-powered with the batteries being re-charged using renewable energy technologies such as solar panels (Farbod and Todd, 2007; Coates and Delwiche, 2009; Wu et al., 2011; Mannweiler et al., 2012; Johnson, 2013; Rey-Moreno et al., 2013). Thus, techniques to improve the energy-efficiency of rural, battery-powered WMN deployments are worthy of investigation.

Rural deployments employ the Infrastructure Wireless Mesh Network (I-WMN) paradigm where a two-tiered network architecture is employed. This architecture is depicted in Figure 1.1. The upper-tier comprises a decentralised mesh backbone where the routing and self-organisation capabilities reside. Cost considerations in rural deployments often

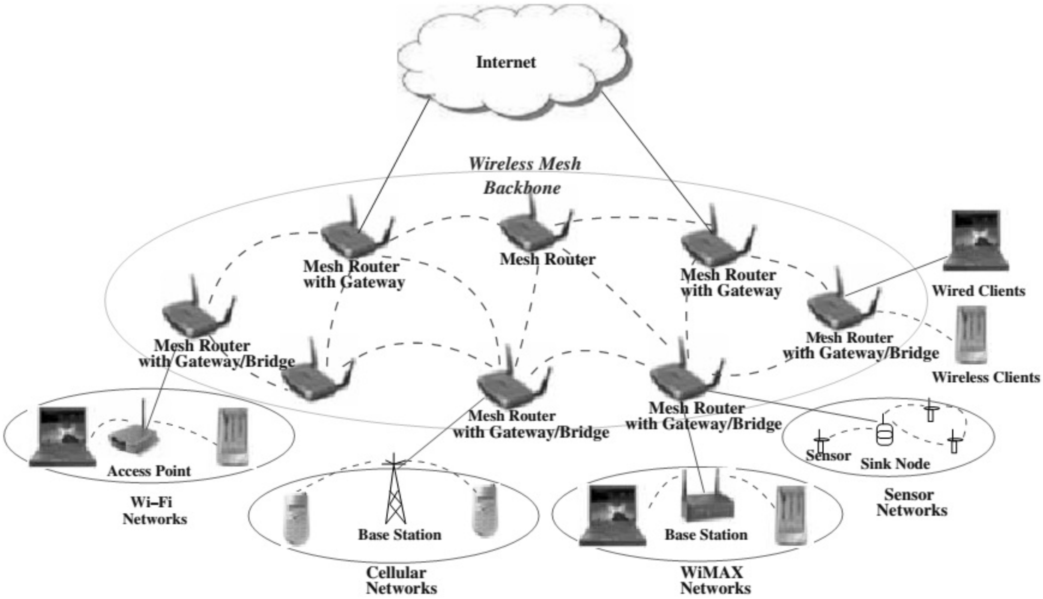
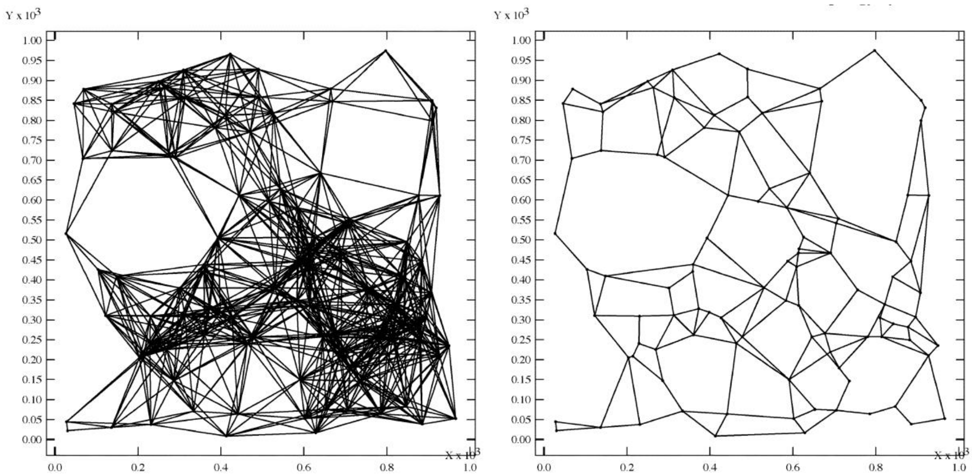



FIGURE 1.1: Infrastructure WMN (Akyildiz and Wang, 2009)

force the use of low-cost, resource-constrained nodes for the mesh backbone (Johnson, 2013). The lower tier comprises client devices that connect to the backbone network.

Topology Control (TC) is a candidate technique for improving the energy-efficiency of rural, battery-powered I-WMN deployments. TC aims to create a sparser network topology by adjusting the transceiver powers of the network nodes (see Figure 1.2). Sparser networks are achieved by minimising the wireless transceiver power output of the network nodes whilst maintaining the desired network connectivity. The wireless transceiver is reported in Santi (2005) to account for between 15% to 35% of the total power consumed by a wireless node. Thus, TC's targeting of the wireless transceiver can have an impact on reducing the total power consumed by an I-WMN node.

The study of TC for wireless ad-hoc networks (including mesh networks) has been largely limited to mathematical models and simulations. Fewer reports of the study of the efficacy of TC prototypes for the I-WMN can be found in the literature. The lack of TC prototypes in general has been regarded as *"probably the most important open issue in this research field"* in Santi (2005). According to Akyildiz and Wang (2009), the reason for this state of affairs is that *"many topology control schemes are proposed without considering the computation complexity or the possibility of being implemented. Although the performance is justified through theoretical analysis, they*



(a) Pre Topology Control (b) Post Topology Control

FIGURE 1.2: Effect of Topology Control (Li et al., 2005b)

*are difficult to be implemented"*. Numerous prototypes for Wireless Sensor Networks have since been developed but prototypes for I-WMNs have not been as forthcoming. The lack of TC prototypes for I-WMNs has meant that the multi-dimensional effectiveness demonstrated by TC schemes in the literature has not yet been experimentally demonstrated in these networks.

The IEEE 802.11s mesh networking standard has been proposed to ensure interoperability between the various vendor hardware devices that can be used in I-WMN deployments. The standard has subsequently undergone an amendment but, according to Wei et al. (2013), better solutions for power management are still required.

Thus, the combination of a lack of studies of TC prototypes for I-WMNs and the need to improve the mesh networking standard's treatment of power management motivates this study. The goal of this thesis is to study the effectiveness of TC as a standards-compatible technique for improving the energy-efficiency of rural, battery-powered I-WMNs by producing transceiver power savings. The study is based upon observations made of, and measurements derived from, a TC prototype implemented on an indoor test-bed.

This thesis describes the design of the PlainTC prototype and subsequently subjects the prototype to a test-bed evaluation. The evaluation comprises of two parts: an evaluation of the dynamic, run-time behaviour of the prototype and a traditional, static evaluation based on performance snapshots at regular intervals.

The dynamic run-time evaluation documents the topology instability caused by the prototype. Frequent changes in nodes' transceiver powers are identified as the cause of the instability and a context-based solution is proposed. This solution produces a 45% reduction in the number of transceiver power changes experienced by the network with improvements to cumulative transceiver power savings and data traffic performance also being recorded.

The key contributions of this thesis are:

- The establishment (via simulations and test-bed evaluations) that the Critical Number of Neighbours (CNN) connectivity strategy is a viable building block of a TC implementation for an I-WMN. Previous studies (Hall, 1985; Philips et al., 1989; Santi and Blough, 2003; Xue and Kumar, 2004; Wan and Yi, 2004; Balister et al., 2005) have established the mathematical basis for creating connected networks using CNNs. This study establishes, via simulations and test-bed evaluations, that the CNN connectivity strategy can be practically employed to maintain network connectivity whilst producing cumulative transceiver power savings.

- An analysis of the design and implementation decisions taken to produce an IEEE 802.11s compatible, autonomous TC scheme suitable for low-cost, resource-constrained I-WMN nodes. The existing I-WMN applicable TC schemes found in Kawadia and Kumar (2005); Valera et al. (2008); Kowalik et al. (2008a) are unsuitable for low-cost, resource-constrained nodes as they require frequent power level switching, introduce additional communication overheads and modify routing tables. In addition, these features result in the schemes being rendered incompatible with the IEEE 802.11s mesh networking standard. This study is the first to document the benefits derived from basing an autonomous TC scheme upon a CNN connectivity strategy. This study also documents the implementation decisions taken to produce a working TC prototype that is compatible with the IEEE 802.11s standard whilst being suitable for low-cost, resource-constrained I-WMN nodes.

- It is well-documented in the literature that TC improves both energy-efficiency and spatial reuse in ad-hoc networks, including WMNs. These findings are largely based upon simulation-based studies whilst previous reports of TC prototypes applicable to the I-WMN have largely focussed on design issues. Therefore, the opportunity presents itself to conduct a prototype-based evaluation on the effectiveness of TC as an energy-efficiency and performance improvement mechanism. This work is the first to document the dynamic real-time behaviour of a TC prototype applicable to the I-WMN and also investigates the energy-efficiency and performance of the network topologies created by the proposed scheme. This study establishes that the energy-efficiency of the I-WMN is negatively impacted even though cumulative transceiver power savings may be achieved and that the position of a node influences its resultant transceiver power output. In addition, the increased path lengths recorded were found to degrade network performance.

- Even though contextual information has been previously employed in WMNs, as found in Kubo et al. (2010); Matos et al. (2012), this study represents the first time that it has been used to reduce the topology instability caused by employing a CNN connectivity strategy in a TC scheme. Transceiver power changes caused by the maintenance of a CNN are shown to exacerbate topology instability as measured via context variables. The Principal Component Analysis (PCA) statistical method is applied to historical data to derive the weights associated with each variable. A threshold value is defined that only permits a node to adjust its transceiver power output if the observed change in the node's context exceeds the threshold. This threshold mechanism is incorporated into the PlainTC+ prototype and is shown to reduce topology instability whilst narrowing the performance gap to the Max. Power scheme.

The remainder of this thesis is arranged as follows. Chapter 2 presents the background concepts and context for this study. Exploratory experiments assessing the viability of the Critical-Number-of-Neighbours-based connectivity strategy as the basis for a TC prototype are contained in Chapter 3. Chapter 4 explores the design and implementation of the PlainTC prototype. Chapter 5 details the evaluation setup and the measurement methodologies employed for the various metrics considered. Chapter 6 contains both simulation and test-bed-based evaluations of the PlainTC prototype. Chapter 7 describes a solution to the topology instability problem observed in the previous chapter. A context-change threshold based upon weighted context variables is proposed. A methodology for determining the associated weights is also presented together with a test-bed evaluation of the enhanced PlainTC+ prototype. Chapter 8 considers the use of the proposed prototypes in Wireless Sensor Network and Mobile Ad-hoc Network scenarios. The Thesis is concluded in Chapter 9.

# Chapter 2

# Background

## 2.1 Introduction

This Chapter introduces a Wireless Mesh Network (WMN) and its applicability as an alternative network infrastructure to fixed-line, ethernet and optical networks. The lower deployment costs associated with WMNs have resulted in many deployments where traditional networking infrastructure has been absent. Rural areas in developing countries may also suffer an absence of grid-based electricity and therefore battery-powered I-WMNs have been reported in the literature. Additional peculiarities of rural, developing world WMN deployments are discussed in Section 2.3. Improving the energy-efficiency of these networks is of research interest.

A particular mechanism for reducing wireless transceiver power output (and improving energy-efficiency) called Topology Control (TC) is subsequently discussed in Section 2.4. Future I-WMN deployments will be subjected to the IEEE 802.11s standard and Section 2.5 discusses the co-existence and compatibility of TC with this standard.

A TC prototype is subject to challenges that do not have to be considered by theoretical proposals. These various challenges are presented in Section 2.6 as the basis of a requirements specification for a TC prototype.

## 2.2 Wireless Mesh Networks

Wireless Mesh Networks (WMNs) are a subclass of ad-hoc networks. WMNs are employed to provide a cheaper and more robust alternative to fixed-line, ethernet and optical networks. Thus, as discussed in Akyildiz and Wang (2009), WMNs are primarily used to provide a networking infrastructure in areas where it would be economically unfeasible to deploy the more traditional forms of networks. WMNs can subsequently be interconnected with any IP-based network and deployments can be scaled as demand increases.

WMNs are designed to be dynamically self-organising and self-configuring. These characteristics ensure the robustness of the network whilst minimising the need for human monitoring. Akyildiz and Wang (2009) shows that WMNs have been employed for usage scenarios as varied as commercial wide-area networks, home networking, community networking and building automation.

There are three (3) predominant wireless technologies that have incorporated mesh networking functionality as described in Wei et al. (2013). These technologies are WiFi, WiMAX and LTE. The standards that define the WiMAX and LTE technologies adopt mesh networking as a cheaper mechanism for extending the coverage of existing base stations. Relay stations (nodes) are deployed to provide the extended coverage and, in some instances, to decentralise network management. The relay stations are connected to the base station and each other with point-to-point high-capacity wireless mesh links in order to offer the expected QoS. This arrangement is depicted in Figure 2.1.

Both WiMAX and LTE networks operate in licensed spectrum bands and require large amounts of investment for a network deployment. In addition, these networks are well-planned and are therefore not truly self-organising.

WiFi-based WMNs operate in the unlicensed ISM bands and are much cheaper to deploy. Thus, the WiFi technology is the predominant WMN deployment technology for home and community networking. These deployment scenarios are largely unplanned

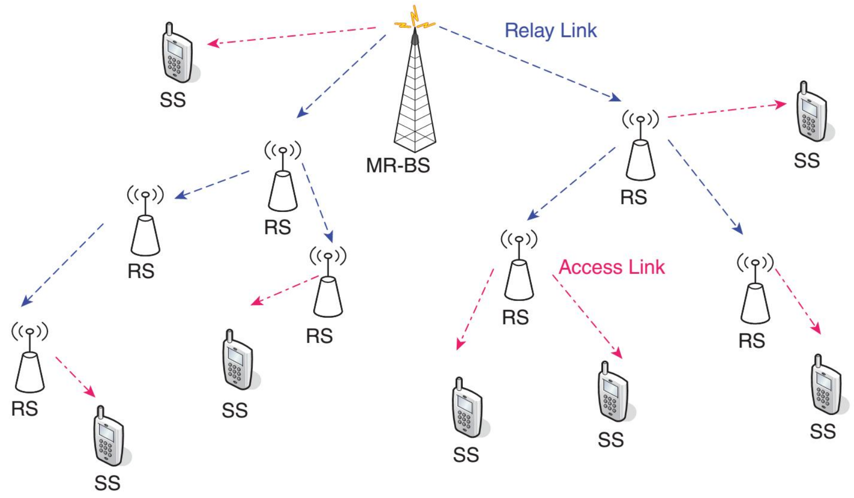



FIGURE 2.1: Example of WiMAX and LTE usage of mesh networking (Wei et al., 2013)

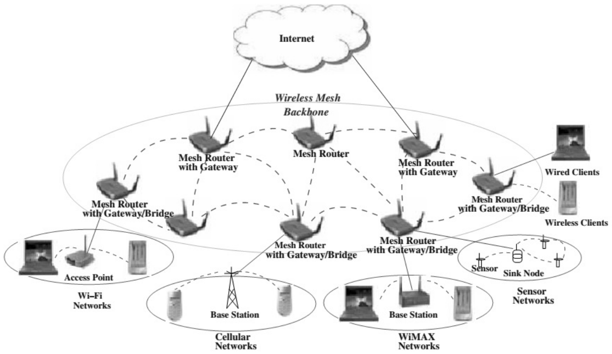
and therefore have the greatest requirement for self-organising and self-configuring solutions. WiFi-based WMNs are thus the focus of this thesis and any future reference to a WMN is assumed to be to a WiFi-based WMN.

Three WMN architectures exist in the literature: Infrastructure WMNs, Client WMNs and Hybrid WMNs. Infrastructure WMNs are used for community networking and incorporates a dedicated backbone of mesh nodes that serve as access points for client devices or other networks (see Figure 2.2(a)). Client WMNs contain client devices that communicate in a peer-to-peer manner and are used for home networking (see Figure 2.2(b)). A Hybrid WMN comprises both a backbone mesh network between dedicated mesh nodes as well as a mesh network between client devices (see Figure 2.2(c)).

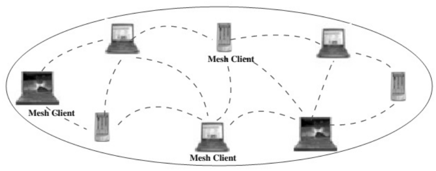
According to Akyildiz and Wang (2009), Infrastructure WMNs (I-WMNs) have been the most common WMN architecture in use. The situation remains unchanged and the I-WMN is thus the focus of this study.

## 2.3 Wireless Mesh Networks for Developing Countries

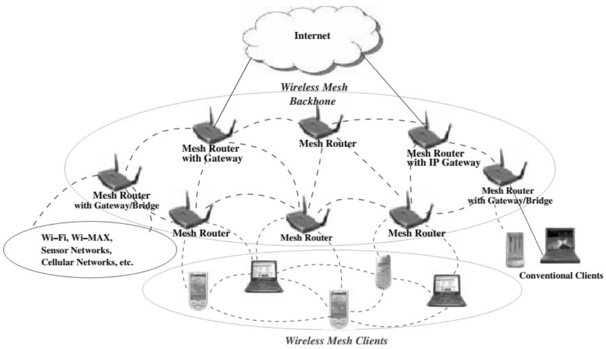
I-WMNs in developing countries are primarily deployed in rural areas as a source of Internet connectivity. These deployments are meant to alleviate the digital divide between




(a) Infrastructure WMN




(b) Client WMN




(c) Hybrid WMN

FIGURE 2.2: WMN Architectures (Akyildiz and Wang, 2009)

rural and urban areas.

Rural I-WMN deployments are cost-sensitive and therefore utilise low-cost hardware and open-source software as enablers. Hardware is often donated at the end of its life cycle and is of a lowly specification. Deployments are largely unplanned with nodes being co-located with existing infrastructure. This means that site surveys and optimisation of node positions are not performed and the network performance may be compromised as a result. In the event of long-distance rural networks, backbone nodes are deployed to specialised sites in order to facilitate line-of-sight connectivity. Sen and Raman (2007) show that such sites are carefully chosen as there is a high cost associated with antenna towers.

Such deployments are cost-sensitive and often rely on low-cost, resource-constrained nodes to create the I-WMN, as discussed in Johnson (2013). Such nodes are typically

WiFi routers that have had their original, proprietary firmware overwritten with free and open-source alternatives. The alternative firmware augments the router's behaviour by adding mesh forwarding and management functionality.

Mannweiler et al. (2012) shows that many rural deployments in developing countries rely on batteries charged by a renewable energy source as there is often a lack of grid-based electricity. This lack of grid-based electricity may be caused by the harshness and remoteness of the terrain, lack of electrical infrastructure or prohibitive cost. In cases where batteries are used to power I-WMN nodes, these batteries are also used to power other devices or appliances. Thus, improving the energy-efficiency of I-WMNs is worth investigating in this rural context given the proliferation of these battery-powered networks. In addition, the results would be relevant to the deployers and maintainers of mains-powered networks that are looking to improve the energy-efficiency of their I-WMNs.

## 2.4 Topology Control

I-WMNs are typically deployed by ensuring that each device in the backbone utilises its maximum transceiver power. The use of the maximum transceiver power aids in maintaining network connectivity and simplifies the initial node configuration. The disadvantages of this approach are the high levels of interference, increased contention for the transmission medium, a reduction in network capacity and potentially unnecessary energy consumption.

TC is thus a candidate for improving the energy-efficiency of battery-powered rural I-WMN deployments in developing countries. In addition to the potential power savings, TC may also be useful in deployment contexts where security measures are necessary. Such contexts are typically associated with urban and campus deployments where potentially sensitive information is transmitted. Eavesdropping may be made more difficult by minimising the transceiver powers utilised by I-WMN nodes. This will force such malicious actors to operate in closer proximity to the mesh backbone node which may serve as a deterrent. Thus TC can be applicable to both rural and urban I-WMN deployments.

The wireless transceiver power output may be seen as a potential target for energy-efficiency measures since the transceiver is reported in Santi (2005) to constitute between 15% and 35% of the total energy consumed by a wireless device. Thus, any reductions in transceiver power output that do not compromise the connectivity of the I-WMN should be welcomed. The potential for power savings at the wireless transceiver serves as motivation for this study to consider Topology Control (TC) since such schemes have a direct influence upon the transceiver power of an I-WMN node.

As a result of the inefficiencies associated with maximum transceiver power output in ad-hoc networks, several TC schemes have been developed with the broad goals of maintaining network connectivity and improving network capacity whilst reducing transceiver power output. TC is not unique in this respect as Stallings (2005) reports that cellular networks have been employing power control techniques to achieve the same goals.

TC schemes can be broadly classified into *gradual-adjustment-based* schemes where node transceiver powers can be a value in $[0, tx\_max]$ or *sleep-based (on-off)* schemes where the node employs the maximum transceiver power but sleeps either at regular intervals or when the wireless transceiver is not required, subject to the constraint that a Connected Dominating Set (CDS) is maintained.

TC is based upon earlier efforts to determine the optimal number of neighbours (or node degree) for nodes to possess. The seminal contribution can be found in (Kleinrock and Silvester, 1978). The authors prescribe that the optimal number of neighbours is six (6) in terms of maximising the network capacity and throughput. It should be noted that the result obtained for the optimal number of neighbours is independent of the total number of nodes in the network.

Hajek (1983) suggests that the transmission range be dynamic and adjusted at the beginning of every transmission. The routing strategy used was to adjust the nodes transmission range in order to reach a neighbouring node in the direction of the intended destination node. This strategy does have one obvious limitation when applied to the low cost, resource constrained nodes that are being considered. This routing strategy requires that the sending node know the direction of the intended destination node, which is currently not possible with the network nodes that are typically used in rural I-WMN deployments. However, the mathematical modelling done in this work showed that the adaptive transmission range strategy resulted in each node having an optimal number of close to three (3) neighbours.

Takagi and Kleinrock (1984) differs from most other work in the field by dealing with the optimal transmission power required to maximise the expected one-hop progress that is made in delivering data to its destination. The optimal transmission radii for nodes that are randomly distributed are calculated for both the ALOHA and Carrier Sense Medium Access (CSMA) protocols. The optimal number of neighbours for networks that used the CSMA MAC was estimated to be close to five (5) and is independent of the total number of nodes in the network.

Philips et al. (1989) established a relationship between the optimal number of neighbours and the area of the plane in which the nodes are distributed. Both lower and upper bounds for the optimal number of neighbours were presented: 2.195 $<$optimal number of neighbours $<$10.526.

Xue and Kumar (2004) developed heuristics that can be applied to any wireless multi-hop network, independent of the total number of the nodes in the network. The heuristics provided assistance in arriving at the lower and upper bounds for the connectivity of the network. The optimal number of neighbours lies somewhere within the bounds proposed. One significant proposal emanating from this research is that when each node is connected to less than 0.0074log(n) nearest neighbours, then the network tends to be disconnected with probability 1 as $n$ increases (where $n$ is the total number of nodes in the network). This amounts to the lower bound on connectivity. The proposed upper bound states that if each node is connected to more than 5.1774log(n) nearest neighbours then the network is connected with probability one as $n$ increases.

The upper bound formulated in Xue and Kumar (2004) was improved upon in Wan and Yi (2004). The authors suggested that if each node in the network was able to connect

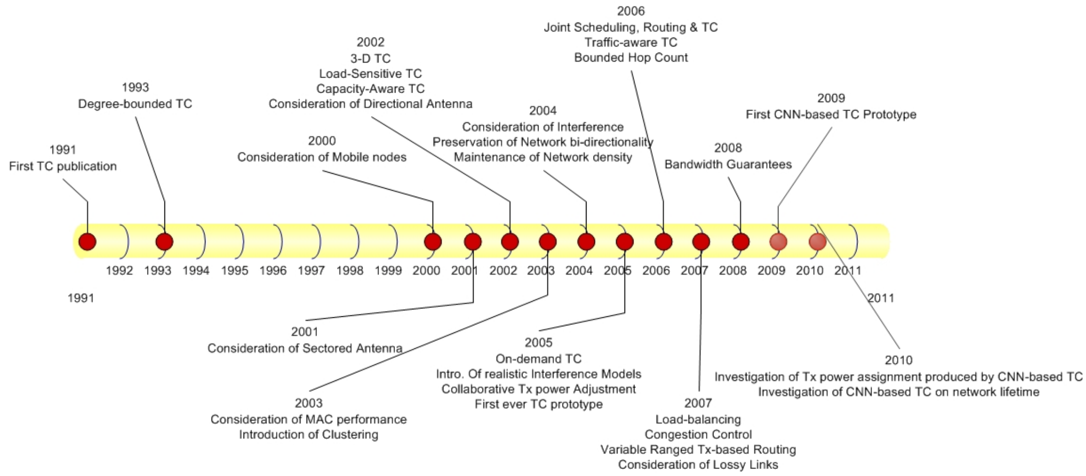



FIGURE 2.3: Timeline of Major Milestones in TC Research

to a maximum of 2.718log(n) nearest neighbours then the resultant network topology would be almost surely connected.

TC schemes were initially proposed to automate the process of maintaining a bounded neighbourhood in wireless ad-hoc networks but later efforts gradually evolved in complexity, see Figure 2.3.

The TC evolution subsequently considered the effects of node mobility and the impact of TC on both single-layer and multi-layer QoS in MANETs and WMNs. Examples of such work can be found in Ramanathan and Rosales-Hain (2000); Huang et al. (2002); Jia et al. (2004); Tang et al. (2005); Kawadia and Kumar (2005); Li et al. (2005b,a); Hu et al. (2006); Siripongwutikorn and Thipakorn (2008); Aron et al. (2008c); Jang and Fang (2008); Valera et al. (2008); Kowalik et al. (2008a). Later works largely focused upon the application of sleep-based TC for WSNs due to the proliferation of sensor devices and the burgeoning Internet-of-Things research domain as can be found in Donglan (2015); Faheem et al. (2015); Roslin (2015); Zhang et al. (2015).

Three trends can be observed from the evolution of TC schemes depicted in Figure 2.3. Firstly, the complexity of the proposed schemes has been increasing over time. Initial schemes focused upon topology construction and maintenance by maintaining bounded node degrees whilst later schemes added QoS considerations. Secondly, the scarcity of TC prototypes that can be used as the basis for further study in MANETs and WMNs is observed. Initial studies were based on mathematical and simulation models of MANETs and WMNs whilst the majority of prototypes were later developed for WSNs. Thirdly, there has been a migration from considering MANETs and WMNs towards WSNs.

It can be argued that the move towards proposing TC schemes for WSNs is a result of the combination of two contrasting factors: the difficulty in translating theoretical TC proposals into functional TC prototypes suitable for MANETs and WMNs that was expressed in Santi (2005) and the surge in Wireless Sensor Network and Internet-of-Things research interests accompanied by the relative ease of prototype development for wireless sensor hardware platforms.

Despite the trend to propose sleep-based TC for WSNs, gradual-adjustment-based schemes remain appropriate for use in I-WMNs. Such schemes can complement existing QoS mechanisms employed in I-WMNs whilst potentially providing cumulative transceiver power savings in these networks. Typical I-WMN usage scenarios discussed in Akyildiz and Wang (2005) require that all backbone nodes remain available, thus rendering sleep-based TC schemes[1] inappropriate. This Thesis therefore focuses only on gradual-adjustment-based schemes which do not utilise sleep states.

The majority of TC researchers have abstracted the MANET and I-WMN topology as a graph and have proposed schemes that rely upon the application of Graph Theory to create optimal sub-graphs. These sub-graphs have been obtained using various graph optimisation techniques.

The most common graph optimisation techniques used in the TC literature are *MSTs* found in Ang and Tham (2008); Aron et al. (2008c); Liu et al. (2008); Namboodiri et al. (2008); Wang et al. (2006); Li et al. (2004); Cardei et al. (2004); Li et al. (2003); Cheng et al. (2003); Ramanathan and Rosales-Hain (2000), the *Forests* found in Wu and Liao (2006); Burkhart et al. (2004), *RNGs* found in Borbash and Jennings (2002); Jennings and Okino (2002) and the *GGs* found in Song et al. (2006, 2004); Rodoplu and Meng (1999).

[1]Such schemes find more relevance in Wireless Sensor Networks

Another common technique applied in the TC literature is the creation of virtual cones using angles such as those found in Bahramgiri et al. (2002) and in Li et al. (2005a). This technique divides a node's transmission range into equal-sized cones and attempts to maintain at least one neighbour within each cone.

The use of Graph-Theory-based techniques in TC schemes possess the advantage of being provable but it is no coincidence that, to the best of this study's knowledge and a source of frustration expressed in both Santi (2005) and Akyildiz and Wang (2009), none of these techniques have yet been prototyped. The reasons for this situation are varied and may not extend to all of the four common Graph Theory optimisation techniques. These reasons are shown below:

- computationally complex algorithms
- unrealistic information requirements
- uniform node density and distribution assumptions
- idealised network and topology conditions
- no consideration of practical node limitations

Graph-Theory-based approaches often require complex, computationally intensive operations to create the desired communication/network sub-graph as noted in Akyildiz and Wang (2009). These approaches usually require the exchange of detailed connectivity information amongst network nodes. Global message exchanges incur significant communication overhead, which is often considered an unrealistic information requirement in distributed networks and systems.

Uniform node densities and distributions are often assumed, in conjunction with idealised communication channels. These assumptions often do not hold in real-world I-WMN deployments, as described in Allen et al. (2005). Lastly, Graph-Theory-based approaches to TC usually assume a fine-grained control of transceiver power output in order to maintain a specified CNN or links to specific neighbouring nodes. The reality is quite different. Typical COTS I-WMN nodes usually possess a finite number of pre-defined power levels that constrain transceiver power output[2]. The lack of fine-grained transceiver power control makes the accurate maintenance of a specific CNN or a specific link extremely difficult.

The Graph-Theory-based schemes have almost universally been evaluated via theoretical analysis and simulations. These evaluation methods have mostly been employed to analyse the properties of the network topologies being created and maintained and in the instances where such evaluations investigate the QoS capabilities of the topologies created, TC is largely depicted as a very effective QoS mechanism.

Studies depicting the positive impact of TC on Application Layer performance can be found in Ramanathan and Rosales-Hain (2000); Huang et al. (2002); Jia et al. (2004); Tang et al. (2005); Li et al. (2005a,b); Hu et al. (2006); Siripongwutikorn and Thipakorn (2008); Aron et al. (2008c); Jang and Fang (2008). The *CONNECT* and *BICONN* schemes described in Ramanathan and Rosales-Hain (2000) were found to produce up to a 227% improvement in throughput compared to the usage of maximum transceiver powers at all nodes.

Both single-path and multi-path intra-mesh traffic scenarios are considered in Jia et al. (2004). Traffic demands were assumed to be known in advance and transceiver powers were assigned such that node loads were maximised, thus aiding load-balancing and improving network capacity.

Multi-channel WMNs were considered in Tang et al. (2005). An interference-aware TC scheme was coupled with a bandwidth-aware routing protocol and the resultant combination produced a 57% improvement in the number of Application Layer data streams in intra-mesh traffic scenarios.

The scheme in Li et al. (2005b) attempts to maintain at least one neighbour in each sector of the node's transmission range. This scheme (*CBTC*) was found to improve network throughput by 144% whilst an optimised version of the scheme improved throughput by 294%. In addition, *CBTC* reduced latency by approximately 53% whilst the

[2]An appraisal of the power control capabilities of a popular I-WMN node can be found in Chapter Three (Section 3.3)

optimised version achieved a 37% reduction when compared to maximum transceiver power usage.

The *LMST* scheme (Li et al., 2005a) improved the total number of Application Layer data packets delivered whilst also improving the energy efficiency of the data transfer. *LMST* was also found to reduce packet retransmissions.

The *TAP* scheme (Hu et al., 2006) was found to produce a 4% improvement in packet delivery compared to the maximum transceiver usage scenario whilst much more substantial improvements in throughput were recorded in Siripongwutikorn and Thipakorn (2008).

The *ABD* and *PRD* schemes (Siripongwutikorn and Thipakorn, 2008) increased throughput from 100Kbps (at maximum transceiver power) to 425Kbps when nodes were stationary.

The *LM-SPT* scheme described in Aron et al. (2008c) was able to match the throughput achieved when utilising maximum transceiver power output whilst *LM-SPT*'s variants improved throughput even further in intra-mesh traffic scenarios.

The *DT+SD* scheme in Jang and Fang (2008) was evaluated using intra-mesh traffic and improved throughput by 152%, lowered packet loss by 36% and lowered delay by 58% when compared to maximum transceiver power usage.

The schemes (*Scheme-1* and *Scheme-2*) reported in Huang et al. (2002) were evaluated in ad-hoc networks possessing directional antennas. For short-range traffic (one-hop), both schemes were reported to reduce end-to-end delay by more than 50%. Improvements in packet loss ratio and throughput were also described. The work in Huang et al. (2002), however, also depicted a scenario where Application Layer traffic performance was reduced. For long-range traffic (10 hops), both *Scheme-1* and *Scheme-2* were found to increase end-to-end delay and degrade throughput when compared to the maximum transceiver power usage. Other reports of performance degradation can be found in Park and Sivakumar (2002) and Mir et al. (2006).

Sole reliance on analytically-derived and simulated TC schemes has provided an idealised version of the efficacy of TC in wireless ad-hoc networks. These schemes have not easily resulted in TC prototypes, forcing researchers to adopt simpler, more practical approaches in order to study the efficacy of TC in real-world wireless ad-hoc networks. The study of the effect of TC on other QoS mechanisms resident within the protocol stack also demands TC prototyping.

To the best of our knowledge, only five TC prototypes that can be considered for use in an I-WMN appear in the literature[3]. Kawadia and Kumar (2005) prototyped the *COMPOW*, *CLUSTERPOW* and *MINPOW* schemes. These schemes create routing tables for each specific power level supported by the wireless cards employed. Each scheme requires the modification of existing packet headers as well as the routing protocol messages, thus resulting in tight coupling between the schemes and the protocol stack. The evaluations of these schemes was limited only to the correctness of the routing tables created since the authors reported that the hardware used crashed repeatedly.

The scheme reported in Valera et al. (2008) required the pre-determination and subsequent manual tuning of a common transceiver power level, on all network nodes, to ensure network connectivity. This resulted in a completely static scheme that lacked scalability. The scheme was found to have a negative effect on a Network Layer routing protocol since the protocol performed best when the nodes operated at maximum transceiver power. The evaluation of the scheme was based on a string topology, which is an uncommon network topology for most I-WMN deployment scenarios.

Kowalik et al. (2008a) presented the *ConTPC* power control scheme that attempts to minimise packet loss caused by transceiver power reductions. A transceiver power reduction is only allowed if the reduction would not affect the delivery rate of the wireless link involved. This method ensures that only high-quality links are considered for transceiver power reductions. The mechanism for computing the delivery rate requires that probe packets be periodically sent at every permissible power level. The disadvantages of such an approach are that it adds to the existing communications overhead and

[3]Numerous sleep-based TC prototypes for WSNs can be found in the literature.

may result in unstable network topologies as node power levels are continuously changing in order to send these probe messages at the various power levels. In addition, the sending and receipt of these probe messages are subjected to the prevailing scheduling mechanism which may result in a delayed view of network conditions.

The TC prototypes discussed above have yet to overcome all of the challenges[4] to be faced in WMN deployments.

## 2.5 Topology Control and the IEEE 802.11s Standard

The advantages of I-WMNs have led to various networking vendors developing and selling mesh networking solutions. Each of these solutions uses proprietary protocols and thus cross-vendor equipment interoperability is not possible.

IEEE 802.11s is an attempt to ensure the interoperability of WMN-enabled devices by providing a standardised framework for the operation of WMNs. 802.11s specifies a two-tiered architecture as shown in Figure 2.4. The upper-tier, or backbone, comprises a Mesh Portal Point (MPP) and Mesh Points (MP). The MPP serves as the interface between an I-WMN and an external network. The MPP also combines the functionality of a Mesh Point (MP) which acts only as a relay node. Mesh Access Points (MAPs) combine the functionality of both a MP and a traditional access point. The access point functionality allows the lower-tier of client devices to connect to the WMN backbone.

Apart from the network architecture, the IEEE 802.11s framework also specifies, amongst other things:

- how WMNs are established
- eligibility requirements for the nodes that wish to join a network
- the procedure to follow when a new node attempts to join an already established network

[4]These challenges are discussed in Section 2.6

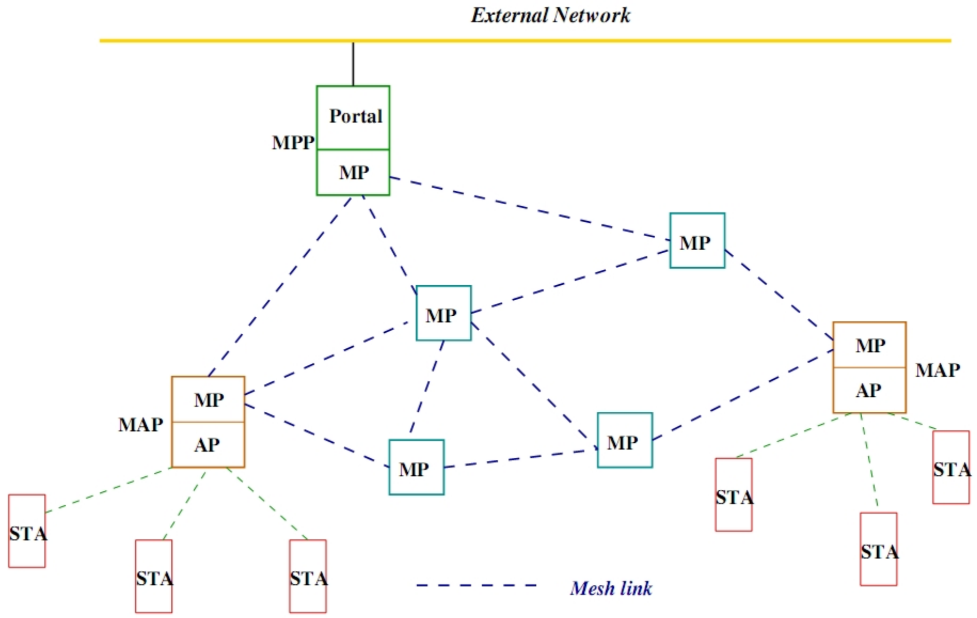



FIGURE 2.4: Common Wireless Mesh Network Architecture (Wang and Lim, 2008)

- the roles that the nodes may play in the network
- the establishment of trusted peer links
- how packet forwarding occurs
- how congestion is controlled
- an option for channel switching in both single- and multi-radio WMN nodes

The IEEE 802.11s standard also attempts to address the energy-efficiency problem by providing an optional power-saving mechanism. This power-saving mechanism allows MPs to sleep and awake at scheduled intervals. The mechanism requires synchronisation of the sleep and active periods as well as the synchronisation and buffering of network traffic. Thus the power saving mechanism is most useful when all of a MP's neighbouring MPs support power saving. A similar power saving mechanism was also a part of the initial IEEE 802.11 standard but its actual implementation in commercial products was found by Anastasi et al. (2008) to be limited due to the additional complexity involved in the synchronisation process, decrease in QoS and increases in energy consumption due to the resultant additional transfer times.

It is therefore highly unlikely that the optional power saving feature of IEEE 802.11s will actually be used in I-WMN deployments. This ultimately means that IEEE 802.11s does not possess a practical power saving mechanism that can improve the energy-efficiency of the battery-powered I-WMN networks that are typically found in the rural areas of developing countries. The opportunity for realising power savings is however not lost. Topology Control (TC) can be applied to the I-WMN to increase the energy-efficiency of IEEE 802.11s-based deployments but existing TC schemes have not been designed to be explicitly compatible with this standard.

A fully distributed TC scheme avoids the complexity of having to synchronise the sleep cycles of backbone nodes as the power consumption of the node is reduced by another means. TC can minimise the transceiver power being employed by a node to a level that is sufficient to maintain the connectivity of the node. The minimisation of node transceiver power level has an advantage over the traditional power saving mechanism since the node, whilst awake, would not be forced to operate using its maximum transceiver power level. Minimising the transceiver power levels of the mesh backbone nodes, via TC, also reduces interference and enhances the spatial reuse of the communication medium. Thus, TC could serve as a practical power saving mechanism for IEEE 802.11s networks.

## 2.6 Prototyping Challenges Faced by Topology Control for Wireless Mesh Networks

The poor record of TC prototyping has been regarded as *"probably the most important open issue in this research field"* in Santi (2005). According to Akyildiz and Wang (2009), the reason for this state of affairs is that *"many topology control schemes are proposed without considering the computational complexity or the possibility of being implemented. Although the performance is justified through theoretical analysis, they are difficult to be implemented"*. Numerous prototypes for WSNs have since been developed but prototypes for I-WMNs have not been as forthcoming.

The lack of TC prototypes for I-WMNs has meant that the multi-dimensional effectiveness demonstrated by TC schemes in the literature has not yet been experimentally demonstrated in these networks. This section discusses some of the challenges that need to be overcome before TC schemes can be implemented on deployed I-WMNs, the real-world effectiveness of such TC schemes can be determined, and the impact of TC on energy-efficiency and network performance can be arrived at.

### 2.6.1 Computational Complexity

Graph Theory easily lends itself to the modelling of wireless ad-hoc networks such as WMNs. Several optimisation techniques derived from decades of effort are then applied to enhance various properties of the network graph. These optimisation techniques are known to be computationally complex and require powerful desktop or server machines for processing.

Typical I-WMN deployments in the developing world[5] use low-cost, resource-constrained nodes for the network backbone. In addition, the open-source firmware resident on such nodes is not optimised for performance. Thus, high-complexity mechanisms are unlikely to execute timeously, if at all.

In addition to the complexity of the graph optimisation logic employed, graph-theory-based schemes have substantial information requirements that may be difficult for a resource-constrained node to process and store for further use. Simpler mechanisms with lower information requirements are needed for TC prototypes. Existing networking mechanisms that either collect or produce information, such as network allocation vectors and routing tables should be exploited in this regard. An approach that leverages existing data sources is demonstrated in Chapter 3 (Section 3.4) and in Chapter 4 (Section 4.6).

[5]previously discussed in Section 2.3

### 2.6.2 Feasibility of Power Control in COTS Hardware

Power control refers to the ability of a node to adjust its transceiver power and is thus a pre-requisite for TC. Studies establishing the ability of commercial-off-the-shelf (COTS) wireless cards to provide transceiver power control have been conducted in Kawadia and Kumar (2005) and Kowalik et al. (2008b). Unfortunately, COTS wireless cards are not typical I-WMN nodes. Thus, an attempt has been made to establish, using a similar methodology to the one employed in Kowalik et al. (2008b), whether the Linksys WRT54GL router, a popular COTS I-WMN node that is considered in this study, was also capable of transceiver power control. In addition, the latencies involved when changing power levels were determined. The results can be found in Chapter 3 (Section 3.3).

### 2.6.3 Maintaining Network Connectivity

One of the most important properties of a TC scheme is its ability to maintain network connectivity. Connectivity maintenance is thus a major contributor to the computational complexity of TC schemes. There are various methods that can be applied to maintaining network connectivity in an I-WMN, such as the use of the MST, CDS, CTR, CNN and the use of maximum transceiver power. The requirement that TC schemes be of low complexity, espoused in Akyildiz and Wang (2009) rules out the MST and CDS methodologies whilst the requirement for distributed TC schemes rules out the CTR methodology. The requirement for producing transceiver power savings rules out the use of the maximum power assignment method. Thus, the CNN approach appears most suitable for use in a low-complexity, practical TC scheme for the I-WMN.

The CNN connectivity approach has been extensively explored in the literature, as found in Kleinrock and Silvester (1978); Hajek (1983); Takagi and Kleinrock (1984); Hall (1985); Philips et al. (1989); Blough et al. (2003b); Xue and Kumar (2004); Wan and Yi (2004) and is concerned with maintaining the minimum number of neighbours that is required to ensure network connectivity. This approach is low in complexity, can be determined in a distributed manner and results in heterogeneous transmission power (and range) assignment as long as the number of neighbours prescribed by the CNN is maintained. In addition, the CNN is less-affected by the distribution of the network nodes, so there is no need to assume a uniform or homogeneous I-WMN node distribution. A further examination of the connectivity capabilities of various CNN schemes can be found in Chapter 3 (Section 3.2).

### 2.6.4 Non-uniform Backbone Node Distribution

Analytical evaluations and simulations of TC schemes often assume a uniform distribution of network nodes. This assumption rarely holds in deployed I-WMNs as the backbone nodes are typically co-located with existing infrastructure such as houses and other buildings. These objects are not uniformly distributed, resulting in non-uniform distribution of backbone nodes when deployed.

The non-uniform distribution of network nodes results in non-uniform network topologies, which often require maximum transceiver power output to maintain network connectivity. A closer investigation of how node distribution affects transceiver power savings can be found in Chapter 3 (Section 3.2).

### 2.6.5 Compliance with IEEE 802.11s

IEEE 802.11s (IEEE, 2008) refers to a standard that contains the specification for WMN operations. Amongst other things, 802.11s defines QoS mechanisms such as routing, congestion control, power management, security and hand-off. TC schemes must be designed so as not to counteract the effects of these QoS mechanisms, thus ensuring a successful co-existence. An analysis of the standards-compatibility of the prototyped TC scheme presented in this thesis can be found in Chapter 4 (Section 4.4).

### 2.6.6 Maintaining Topology Stability

I-WMNs naturally experience topology instability due to the transient nature of wireless links when deployed. Such instability was observed in Allen et al. (2005); Lundgren et al. (2006); Camp et al. (2006). TC schemes possess the tendency to exacerbate this instability when node transceiver power adjustment occurs. Topology instability may be an unintended side-effect of the use of a CNN-based connectivity strategy by a TC scheme as the need to maintain the required number of neighbours may cause constant transceiver power fluctuations, thereby causing topology instability.

Changes in transceiver power can cause fluctuations in link quality. These fluctuations could, in turn, cause route fluctuations or flapping subject to the use of a link-quality-based routing metric, impacting negatively on network performance. Topology instability caused by a TC scheme must be minimised, or if possible, completely eradicated. A mechanism for reducing the instability caused by TC is presented in Chapter 7.

### 2.6.7 Coupling within the protocol stack

Coupling is a software engineering term that relates to the dependencies that may exist between two or more software components. In the context of this thesis, coupling refers to the level of integration between the TC scheme (a software component) and the pre-existing protocol stack (a collection of software components).

Loose-coupling is preferred to tight-coupling as modularity and extensibility are promoted. These two software engineering architectural features can help to better integrate a TC scheme into the protocol stack without affecting the operation of existing stack protocols. In addition, future enhancements to the operating logic of a TC scheme will not result in subsequent modifications to the traditional stack protocols. Thus, a "sandbox effect" is created.

A discussion of the architectural integration of the proposed TC scheme with the traditional protocol stack can be found in Chapter 4 (Section 4.5).

### 2.6.8 Automating Topology Control

For TC schemes to enjoy wide-spread usage in I-WMN deployments, the execution of such schemes must be transparent to the end-user. Minimal or no user setup and interaction should be required (especially if rural African areas, which are characterised by a lack of technical expertise, are considered). In addition, the dynamic nature of the I-WMNs and the network topology when deployed (despite usually comprising stationary nodes) lends itself to the use of autonomous mechanisms. In the context of this thesis, automating TC means that TC schemes need to be able to autonomously react to changes in the network topology in order to maintain optimal network connectivity. An approach to automating the proposed TC prototype is described in Chapter 4 (Section 4.6)

## 2.7 Chapter Summary

This chapter has presented the concept of using a low-cost, low-maintenance network infrastructure such as a I-WMN to bridge the digital divide in developing countries by serving as a low-cost, low-maintenance networking infrastructure. These networks are often deployed in rural areas with unreliable or non-existent electrical infrastructure and must be powered by batteries. Thus, the energy-efficiency of these networks in this particular power-scarce context is an important consideration.

TC has been shown in the literature to produce power savings in ad-hoc networks. In particular, the gradual-transceiver-power-adjustment-based form of TC can be applied to the I-WMN to produce transceiver power savings. The use of gradual-transceiver-power-adjustment-based TC is also compatible with the IEEE 802.11s standard that governs the operation of a WMN and is a viable alternative to the optional power-saving mechanism that the standard proposes.

The scarcity of TC prototypes for I-WMNs is largely a result of the computational complexity with which TC schemes are associated. This chapter considered the challenges to be overcome to ultimately produce a TC implementation that can be successfully operated on a deployed WMN. These challenges can ultimately be considered to be a minimal requirements specification for a working implementation of a TC scheme.

This chapter has briefly argued for the use of CNN strategies as a means of maintaining network connectivity in a TC scheme. The CNN approach can meet several of the challenges described in Section 2.6 and is thus seen as a candidate building block for a prototype.

The next chapter describes exploratory experiments that assess the feasibility of two important building blocks of a TC prototype. The practicability of achieving transceiver power adjustment in a typical I-WMN node is presented. The chapter subsequently investigates the viability of various CNN strategies for maintaining network connectivity and producing transceiver power savings in a WMN.

# Chapter 3

# Exploratory Experiments

## 3.1 Introduction

This chapter presents exploratory experiments undertaken to assess the feasibility of implementing TC in an I-WMN. These experiments were motivated by the analysis of the TC literature provided in Section 2.4 and remarks in Santi (2005) that the poor record of TC prototyping is *"probably the most important open issue in this research field"*. According to Akyildiz and Wang (2009), the reason for this state of affairs is that *"many topology control schemes are proposed without considering the computation complexity or the possibility of being implemented. Although the performance is justified through theoretical analysis, they are difficult to be implemented"*. The scarcity of reports of TC prototypes suitable for use in I-WMNs found in the literature, as shown in Chapter 2, served as additional motivation for these exploratory experiments and the methodology employed.

The feasibility experiments are presented in three (3) phases, beginning with simulation-based experiments and ending with test-bed-based experiments. The connectivity strategies described in Section 3.2 are assessed, via simulation, to determine their potential to produce transceiver power savings whilst maintaining network connectivity in an I-WMN.

Section 3.3 of this chapter then demonstrates the feasibility of programatically adjusting the transceiver power level of a popular, low-cost I-WMN node.

Lastly, the connectivity strategies that were initially assessed via simulation are evaluated on an indoor I-WMN testbed. The initial simulation-based experiments are repeated on the test-bed and can be found in Section 3.4.

The experiments contained in Sections 3.2, 3.3 and 3.4 extend existing knowledge. The simulation-based evaluation found in Section 3.2 differs from those in the literature, such as in Wan and Yi (2004); Xue and Kumar (2004); Balister et al. (2005), by quantifying the cumulative transceiver power savings that are produced by the various connectivity strategies as well as in identifying that nodes closer to the network centre will contribute the majority of the power savings achieved. The experiments contained in Section 3.3 established that the popular Linksys WRT54GL node is capable of dynamic transceiver power adjustment and is thus suitable as a hardware platform for a TC prototype. Existing studies have evaluated Critical-Number-of-Neighbour-based connectivity strategies via simulation, as found in Santi and Blough (2003); Wan and Yi (2004); Xue and Kumar (2004); Balister et al. (2005). Section 3.4 contains experiments that take advantage of the Linksys node's newly-established capabilities in order to subject CNN-based connectivity strategies to a test-bed-based evaluation. The findings in this Section corroborate the findings of Section 3.2 and establish that the CNN connectivity strategy is a viable building-block of a TC prototype for an I-WMN.

## 3.2 Feasibility of using Connectivity Strategies to produce transceiver power savings in Wireless Mesh Networks

Connectivity in I-WMNs can be achieved by using one of three possible approaches. The first approach specifies that each node utilises its maximum transceiver power. The second approach determines the minimum transmission range, also dubbed the Critical Transmitting Range (CTR), required to maintain a connected network and the transceiver power output for all nodes is adjusted to sustain this transmission range. The third approach determines the optimal number of neighbours to be maintained in order to ensure network connectivity.

The CTR approach results in the homogeneous assignment of transceiver powers and may not minimise the total transceiver power output. Kawadia and Kumar (2005) indicates that this approach is highly susceptible to the effect of outlying nodes that force a high common power level (or equivalently, transmission range). Using the CTR to achieve network connectivity may be done in one of three ways, each with its own disadvantages. The first technique requires that a central node determines the appropriate CTR and this value is subsequently broadcast throughout the network. Each node then automatically adjusts its own transceiver power output. The second technique also requires that the CTR is determined at a central location but the network nodes are manually adjusted to maintain this transmission range, as described in Valera et al. (2008). The third technique requires that all nodes broadcast their positions and the CTR is subsequently determined locally at each node, generating high messaging overheads. Thus, the practicality of the use of the CTR approach becomes limited when mobile nodes or dynamic network sizes are taken into account.

An alternative approach is for each node to maintain an optimal number of one-hop neighbours, also referred to as the Critical Number of Neighbours (CNN). This approach may result in heterogeneous transceiver power outputs, potentially maximising transceiver power savings. In addition, the CNN is less affected by the distribution and position of network nodes so there is no need to assume a uniform or homogeneous backbone node distribution or a GPS-enabled node. Lastly, maintaining connectivity via a CNN potentially eliminates human intervention (especially when a proactive routing protocol is employed) which is of fundamental importance if true autonomous configuration is to be realised in I-WMNs.

The CNN approach possesses the advantage of being distributed in nature and reliant on locally-available information. This approach is also the most likely to lead to autonomous TC mechanisms that are able to produce cumulative transceiver power savings whilst maintaining network connectivity. Thus, the CNN approach forms the basis for the experimentation reported in this chapter.

Prior research has produced CNN values that are fixed and those that adapt to the network size (total number of nodes), which are discussed below.

### 3.2.1 Fixed CNN

The work in Kleinrock and Silvester (1978) proposed a CNN of 6 which was later adjusted to 8 in Takagi and Kleinrock (1984). The work in Hajek (1983) suggested that the transmission range be dynamic and adjusted at the beginning of every transmission and the modelling of the adaptive transmission strategy resulted in each node having an optimal CNN of 3. CNN values of 8 and 6 were also proposed in Hou and Li (1986). It must be noted that these constant CNN values were derived for the optimisation of packet forwarding strategies and that network connectivity was not explicitly considered.

Works that have taken network connectivity into account have also produced size-independent CNN values. In particular, Blough et al. (2003a) showed that the CNN converges to 9 as the network size approaches $\infty$. This result was shown to hold when the connectivity requirement was relaxed such that at least 95% of the nodes found themselves in the giant component of the original network. The work described in Karyotis et al. (2010) proposed an average CNN of 6 for both non-uniform and arbitrary node distributions.

### 3.2.2 Adaptive CNN

Works taking network connectivity into account have also provided ranges within which the CNN can be found. One of the earliest of such works was described in Hall (1985) which proposed that the CNN could be found within the range expressed in (3.1),

$$2.186 < CNN < 10.588 \tag{3.1}$$

Subsequent research has seen a continuous tightening of the upper- and lower-bounds, firstly in Philips et al. (1989) and subsequently in Xue and Kumar (2004). The work described in Xue and Kumar (2004) found that the network is disconnected with increasing probability as $n$ increases if each node is connected to less than $0.074\log(n)$ nearest neighbours and that the network is connected with increasing probability as $n$ increases if each node is connected to more than $5.1774\log(n)$ nearest neighbours,where $n$ refers to the number of network nodes. This result was shown in Santi and Blough (2003) to be valid for square deployment regions containing both sparse and dense ad-hoc networks.

A further tightening of the upper-bound derived in Xue and Kumar (2004) was obtained in Wan and Yi (2004) resulting in connectivity being assured with high probability if a maximum of $2.718\log(n)$ neighbours are maintained (shown in 3.2),

$$CNN < \alpha 2.718\log(n) \tag{3.2}$$

where any real number $\alpha > 1$ and n is the number of nodes in the backbone network and provided that $n < 10000$. Experiments conducted in Xue and Kumar (2004) suggest that the critical value of $\alpha$ may be close to 1. The connectivity bounds were further improved in Balister et al. (2005) for both directed and undirected graphs, but it has been shown in Santi (2005) that the only way to guarantee full network connectivity in an ad-hoc network is to ensure worst-case connectivity where each node is connected to every other network node.

These size-dependent CNN-based connectivity strategies, except for the worst-case strategy, have been shown to create connected networks with increasing probability as the network size increases to infinity but to the best of our knowledge, their effectiveness in producing transceiver power savings (which is a key motivator for TC) has not been previously evaluated.

### 3.2.3 Experimental Results

CNN connectivity approaches are better suited to I-WMNs with the observed evolution from CNNs that are independent of the network size to CNNs that are derived from the number of network nodes. The connectivity strategies defined in Xue and Kumar (2004), Wan and Yi (2004) and Blough et al. (2003a) were chosen for evaluation. The strategies defined in Xue and Kumar (2004) and Wan and Yi (2004) exemplify adaptive CNNs whilst the strategy found in Blough et al. (2003a) is representative of a fixed CNN strategy. These strategies were chosen for their improvements upon earlier strategies. Newer, improved strategies have not been published to the best of our knowledge.

The selected strategies were simulated to assess their ability to produce transceiver power savings whilst maintaining network connectivity. The simulation methodology employed is detailed in Chapter 5.

#### 3.2.3.1 Network Connectivity

The network is fully connected at the maximum transceiver power output level, where the number of source-destination (src-dest) pairs is $n^2 - n$, where $n$ refers to the number of backbone network nodes. Table 3.1 shows that the *Xue, Kumar* strategy was best able to maintain network connectivity as there was little observed difference in the number of available src-dest pairs for all network sizes. The *Xue, Kumar* strategy benefits from maintaining an adaptive CNN that is based on the size of the backbone network.

The *Wan, Yi* strategy also maintains an adaptive CNN but the CNN is approximately half that maintained by the *Xue, Kumar* strategy. Thus, the Wan, Yi strategy does not provide the same degree of connectivity.

The *Blough* strategy maintains a fixed CNN for a large range of network sizes. This strategy is able to maintain full network connectivity for the smaller network sizes, but does not perform as well as the *Xue, Kumar* strategy due to a lower CNN being maintained at the higher network sizes.

TABLE 3.1: Network Connectivity

| Network Size | Src-Dest Pairs (Max. Power) | Src-Dest Pairs (Xue, Kumar) | Src-Dest Pairs (Wan, Yi) | Src-Dest Pairs (Blough) |
|---|---|---|---|---|
| 20 | 380 | 380 | 380 | 380 |
| 40 | 1560 | 1560 | 1543 | 1560 |
| 60 | 3540 | 3539 | 3501 | 3527 |
| 80 | 6320 | 6317 | 6273 | 6303 |
| 100 | 9900 | 9898 | 9847 | 9882 |
| 120 | 14280 | 14277 | 14196 | 14256 |

TABLE 3.2: Percentage Power Savings Achieved

| Network Size | Xue, Kumar | Wan, Yi | Blough |
|---|---|---|---|
| 20 | 33.5 | 48.4 | 11.7 |
| 40 | 34.3 | 46.6 | 17.3 |
| 60 | 35.4 | 45.7 | 21.2 |
| 80 | 33.3 | 46.5 | 26.1 |
| 100 | 34.3 | 44.4 | 33.5 |
| 120 | 34.1 | 47.9 | 38.7 |

#### 3.2.3.2 Transceiver Power Savings

As shown in Figure 3.1, all three (3) connectivity strategies evaluated were found to produce cumulative transceiver power savings. The extent of the transceiver power savings produced can be found in Table 3.2. Table 3.2 shows that the power savings produced by the adaptive CNN strategies (*Wan, Yi* and *Xue, Kumar*) remained fairly constant as the network size increased whilst the fixed-CNN *Blough* strategy produced increases in power savings. The *Wan, Yi* strategy produced the greatest magnitude of transceiver power savings due to the lower CNN being maintained. The *Xue, Kumar* strategy outperformed the *Blough* strategy at the smaller network sizes but the reverse began to occur at the largest network size. This particular phenomenon is explained by the CNNs required to be maintained by the *Xue, Kumar* and *Blough* strategies respectively. At a network size in the range $[20 : 100]$ the CNN maintained by the *Xue, Kumar* strategy is less than that required by the *Blough* strategy. At greater network sizes, due to the fixed nature of the *Blough* strategy, the adaptive *Xue, Kumar* strategy maintains a greater CNN thus increasing the transmission range assigned to the nodes, thereby limiting the magnitude of transceiver power savings produced.

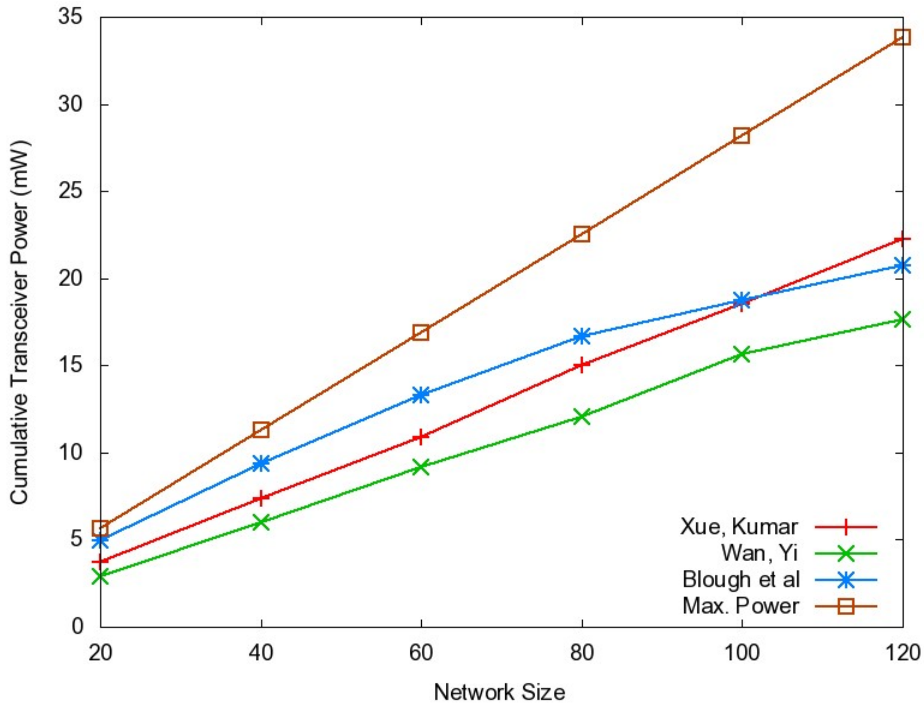


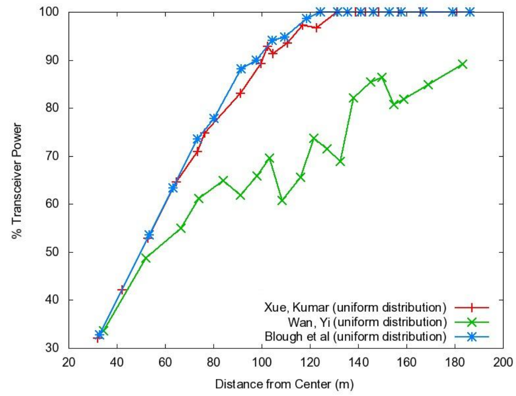

FIGURE 3.1: Cumulative Transceiver Power Output

#### 3.2.3.3 Transceiver Power Assignment

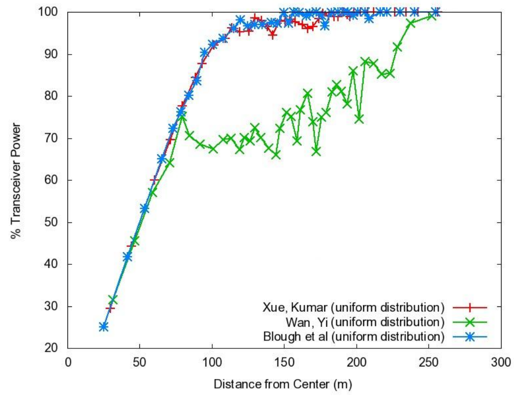
A closer look at the cumulative transceiver power savings produced by all three connectivity strategies reveals an interesting trend. Figure 3.2 depicts the relationship between the transceiver power levels assigned to individual nodes and the distance of these nodes from the (imaginary) network centre.

The evaluation found that the nodes closest to the network centre produced significantly greater transceiver power savings than nodes at the network edge. This phenomenon was observed for all the connectivity strategies and across all the network sizes under evaluation. Edge nodes suffer from situations where fewer candidate neighbours exist and these neighbours are not evenly distributed within the node's transmission range but are rather loosely concentrated in a particular direction.

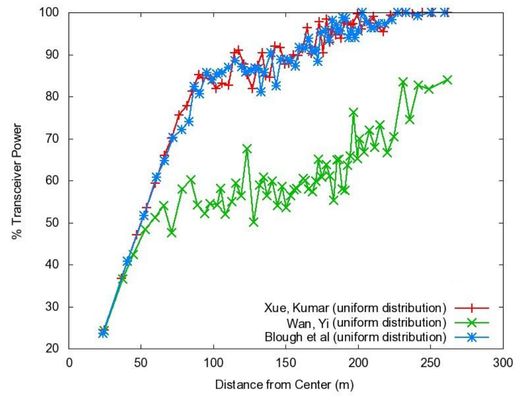
The correlation between the node position relative to the network centre and the resultant transceiver power output is not exclusive to uniformly distributed nodes. This phenomenon is also observed in a clustered environment, see Figure 3.3.



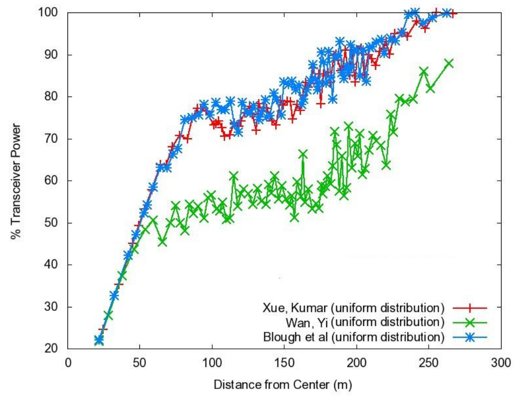

(a) 20 node network



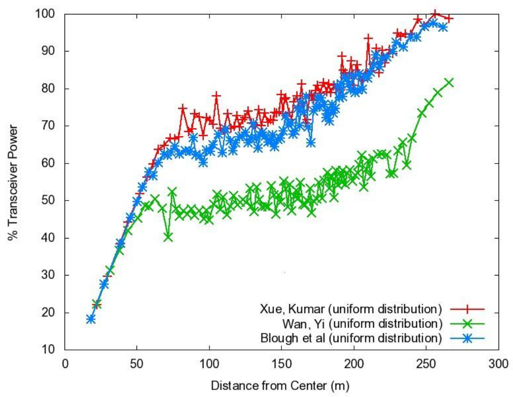

(b) 40 node network



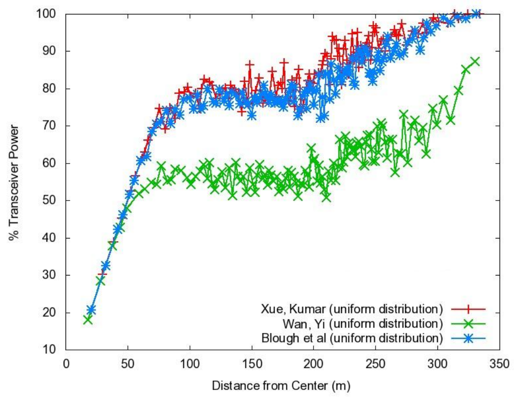

(c) 60 node network




(d) 80 node network

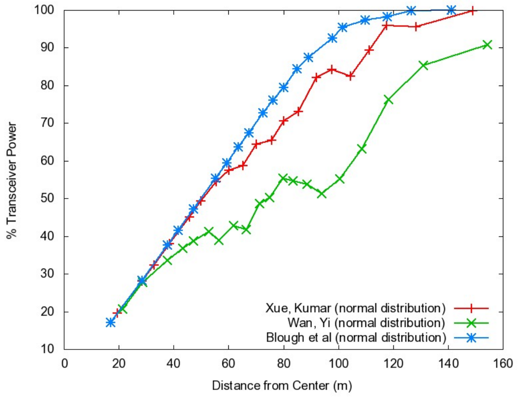



(e) 100 node network

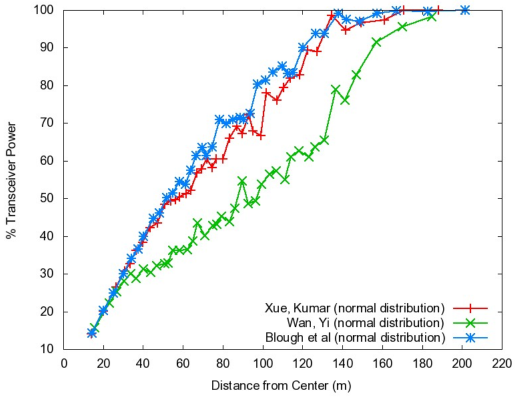



(f) 120 node network

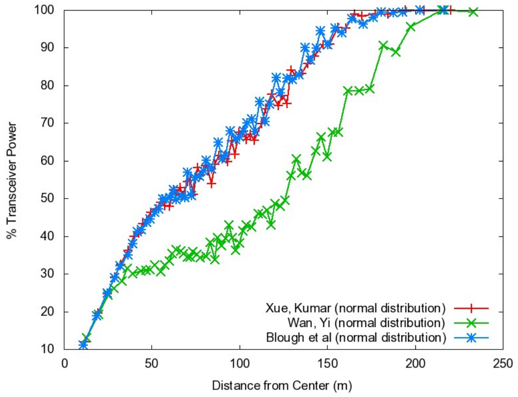
FIGURE 3.2: Transceiver Power Output vs. Distance from Network Center (Uniform Node Distribution)



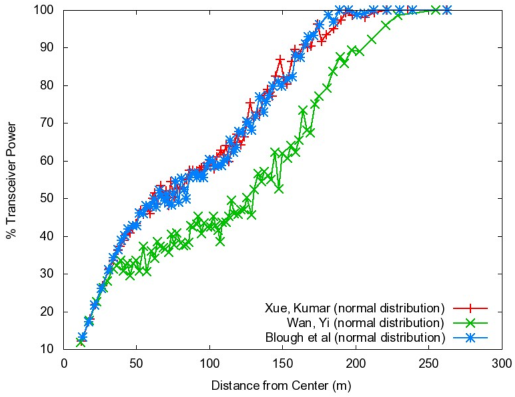

(a) 20 node network



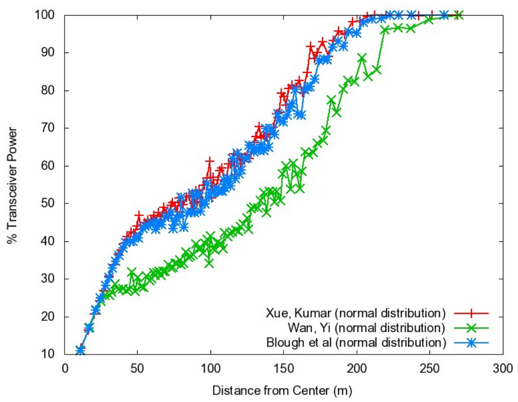

(b) 40 node network



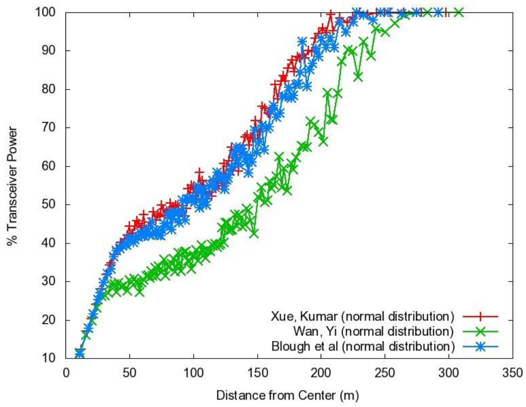

(c) 60 node network




(d) 80 node network




(e) 100 node network




(f) 120 node network

FIGURE 3.3: Transceiver Power Output vs. Distance from Network Center (Normal/-Clustered Node Distribution)

## 3.3 Feasibility of Transceiver Power adjustment in a popular I-WMN node

A study establishing the ability of off-the-shelf wireless cards to provide transceiver power control has previously been conducted in Kowalik et al. (2008b) for Atheros 5212 based chipsets. This chipset is employed in wireless cards that are attached to PCs or laptops but such devices are not utilised as the I-WMN nodes depicted in Figure 2.2 of Chapter 2.

Routers are more appropriate devices for I-WMNS. The Linksys WRT54GL router has been a popular choice for use in I-WMNSs due to its relatively low cost and the availability of open-source firmware for adapting and controlling the node's functionality. Examples of WMN deployments that make use of this device can be found in Lundgren et al. (2006); Bash et al. (2007); Johnson (2007); Freifunk (2008).

The study reported in this Section attempted to establish, using a similar methodology to Kowalik et al. (2008b), whether the Linksys WRT54GL router is capable of transceiver power adjustment since this node makes use of the Broadcom 5352 chipset. In addition, the latencies involved when changing power levels were determined.

### 3.3.1 Ability to change Transceiver Power output levels

The OpenWRT firmware (openwrt.org, 2008) installed on the Linksys WRT54GL router allows for the adjustment of the transceiver power output. The firmware specifies a power output of 19.5dBm by default and after experimentation the use of a 3dBm increment or decrement was adopted. This value was a compromise between the time taken to reach the necessary power level and ensuring that power consumption was minimised.

One Linksys WRT54GL router broadcast frames at 1-second intervals while a laptop was used to capture the frames and log the associated RSSI values. The Linksys router was configured to increase its transceiver power output by 3dBm every two minutes.

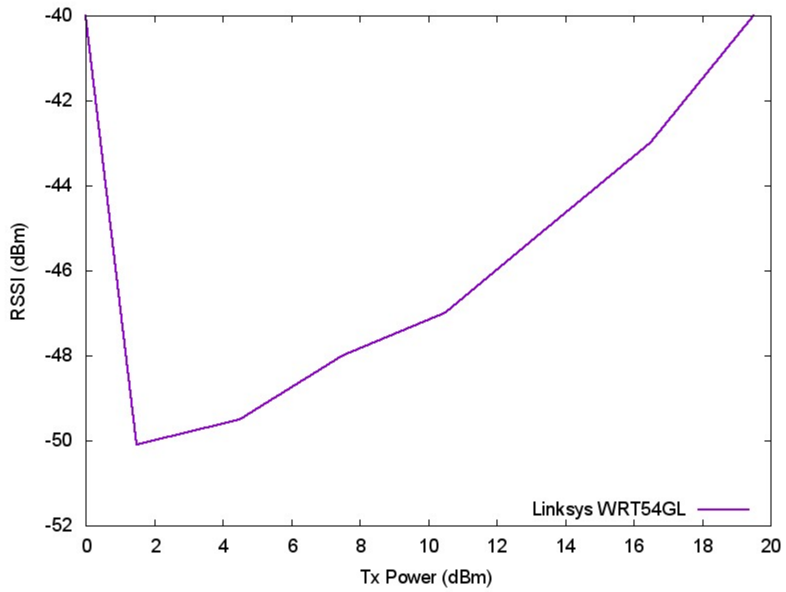



FIGURE 3.4: Received RSSI values for varying transceiver powers

Figure 3.4 depicts the association between the Linksys router's transceiver power output and the RSSI values logged by the laptop. The RSSI values presented are the average of five runs.

It can be observed that the Linksys node exhibits a gradual increase in received RSSI as the transceiver power is increased. Attempting to set the transceiver power to 0dBm proved fruitless as the node automatically reverted to the maximum power.

### 3.3.2 Latency during Transceiver Power Adjustment

The second component of this feasibility study determined the latency involved when changing between transceiver power levels. The same setup used in the previous experiment was employed and the power level was changed every two minutes.

The Linksys WRT54GL router was observed to change transceiver power levels almost instantaneously but required approximately 6 seconds before stabilising at the required level (see Figure 3.5). This particular latency was a result of the process involved when changing power levels. With this particular node, the new power level needed to be stored in the NVRAM (non-volatile random access memory) partition and the wireless

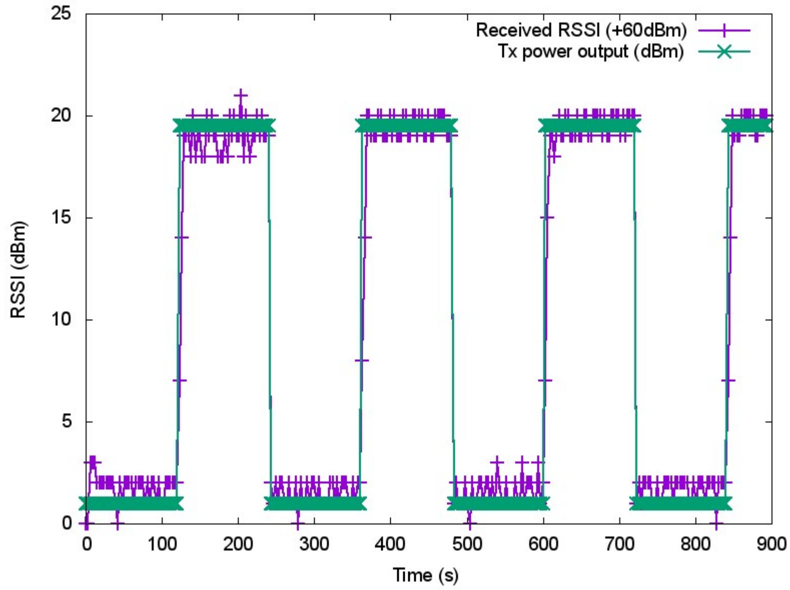



FIGURE 3.5: Transceiver power-change latency

settings needed to be reloaded before the power level change was effected. The RSSI values were observed to stabilise once this process was completed.

The Linksys WRT54GL router is a popular I-WMN node used in deployments around the world (Lundgren et al., 2006; Bash et al., 2007; Johnson, 2007). The Linksys WRT54GL's ability to perform dynamic transceiver power adjustments means that TC schemes are feasible for those WMN deployments that utilise these particular devices as backbone nodes or with other devices that employ the Broadcom 5352 chipset.

## 3.4 Testbed evaluation of Connectivity Strategies

The previous sub-section established the feasibility of dynamically adjusting the transceiver power of a popular I-WMN node. This sub-section investigates the feasibility of employing a CNN-based connectivity strategy as the basis for a TC prototype suitable for low-cost, resource-constrained I-WMN nodes.

The experiments were repeated with the CNN connectivity strategies found in Wan and Yi (2004); Xue and Kumar (2004); Balister et al. (2005). The Worst-Case strategy where nodes are expected to reach all other nodes in one hop and the Max. Power strategy were also considered. A description of the test-bed and the details of the measurement methodology employed can be found in Chapter 5.

### 3.4.1 Transceiver Power Output

This sub-section determines the ability of various CNN strategies to reduce the cumulative transceiver power output of the network. The Max. Power strategy is utilised as a base-line for comparison since this particular strategy is synonymous with an absence of transceiver power adjustment.

Figure 3.6 shows that the Worst-Case CNN connectivity strategy is equivalent to the Max. Power strategy. Thus, no transceiver power savings are produced. The Worst-Case strategy suffers from the requirement that each node be directly connected to all other nodes, resulting in the use of the maximum transceiver power to do so.

The strategy defined in Xue and Kumar (2004) produced transceiver power savings only as the network grew in size whilst the strategies defined in Wan and Yi (2004) and Balister et al. (2005) produced transceiver power savings from the outset. These savings were a result of the smaller number of direct neighbouring nodes to be maintained. It should also be noted that the required CNN to be maintained changed little thus resulting in fairly constant transceiver power consumption. The strategy defined in Balister et al. (2005) for undirected graphs was found to be the best performing strategy by producing the lowest observed cumulative transceiver power output on the test-bed.

### 3.4.2 Network Connectivity

Network connectivity is determined by the availability of paths between all node pairs. The number of node pairs connected using Max. Power was compared to the number of node pairs connected using the various network connectivity strategies (see Fig. 3.7).

The Worst-Case strategy achieved full network connectivity as the maximum cumulative transceiver power was utilised. The strategies defined in Wan and Yi (2004) and in

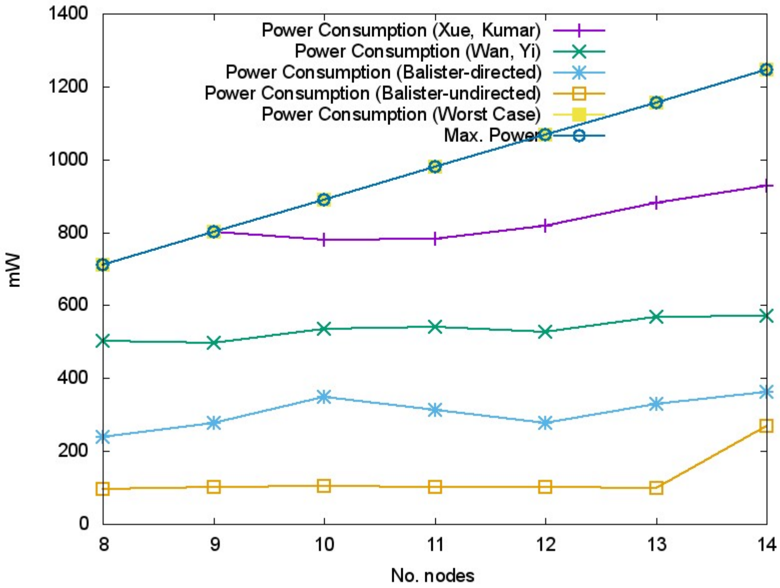



FIGURE 3.6: Cumulative Transceiver Power Consumption

Balister et al. (2005) were not able to maintain a connected network as the CNN upper-bounds employed meant that disjoint clusters were formed and these clusters were not observed to merge as the resultant transceiver power levels were too low.

The strategy defined in Xue and Kumar (2004) was found to maintain network connectivity despite achieving transceiver power savings unlike the strategies defined in Wan and Yi (2004) and in Balister et al. (2005). The Xue and Kumar (2004) strategy benefited from a higher CNN upper-bound which eventually allowed disjoint clusters to converge.

### 3.4.3 Convergence Time

In this sub-section the time taken to establish a fully connected network is discussed. Only the Max. Power, Worst-Case and the strategy in Xue and Kumar (2004) were considered as these were the only strategies to have produced fully connected networks in the previous experiment.

Fig. 3.8 shows that both the Max. Power and Worst-Case strategies show almost no increase in network convergence time as the network size increases. The Worst-Case

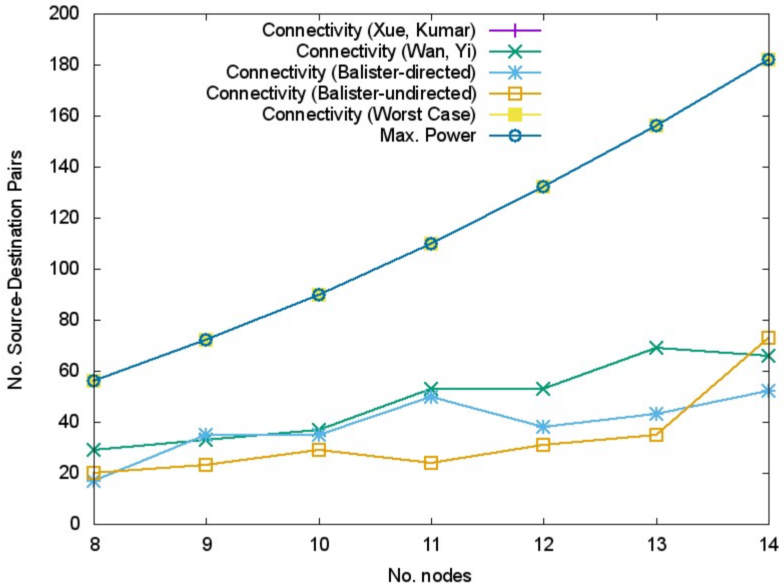



FIGURE 3.7: Network Connectivity

strategy does however incur additional delay due to the need to determine the network size and subsequently maintain the required CNN.

The strategy defined in Xue and Kumar (2004) is shown in Fig. 3.8 to produce an increase in convergence time as the network size increases. This increase can be attributed to the time taken to determine the network size, compute the required CNN and subsequently adjust the transceiver power output to maintain the CNN value. Since this scheme specifies the smallest CNN to be maintained, smaller transceiver powers are required and a maximum number of nodes adjust their transceiver powers. A granularity of 3dB was used to adjust node transceiver powers resulting in multiple adjustments to an individual node's transceiver powers, with each adjustment adding a finite amount of time to the network convergence latency.

## 3.5 Chapter Summary

This chapter has presented experiments assessing the feasibility of developing a TC prototype for the I-WMN. Two important building blocks of a TC prototype were focussed upon: the viability of a CNN connectivity strategy for maintaining network connectivity

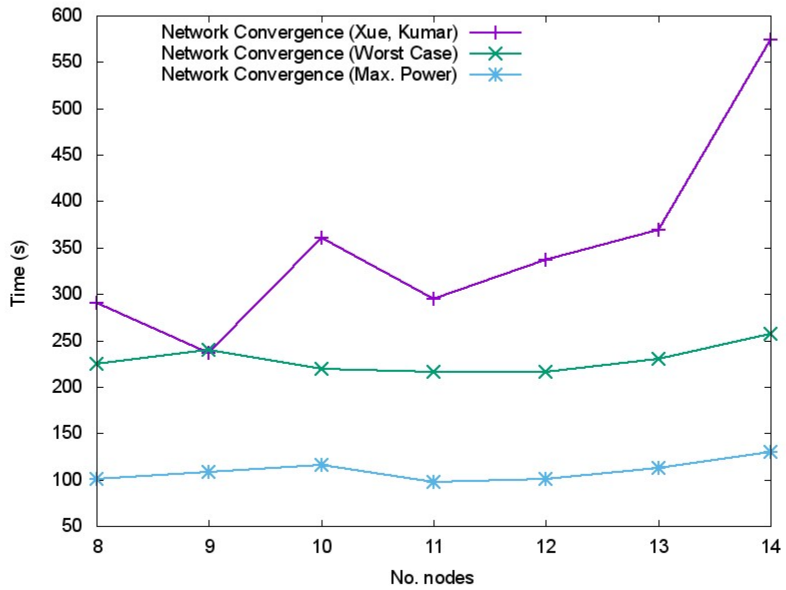



FIGURE 3.8: Network Convergence Time

whilst producing transceiver power savings, and the possibility of achieving transceiver power adjustment in a typical I-WMN node.

The chapter has confirmed that the popular Linksys WRT54GL node used in deployed I-WMN networks is indeed capable of programmatic transceiver power changes. These particular experiments showed that this node is a viable hardware platform for a TC prototype.

This chapter has also addressed the second important building block by showing via simulation- and test-bed-based experiments that a CNN connectivity strategy can be a viable mechanism for use in a TC prototype. The simulation-based study revealed that the node position affects the quantum of transceiver power savings produced whilst the test-bed-based study produced a viable CNN connectivity strategy for use in a TC prototype. The CNN connectivity strategy defined in Xue and Kumar (2004) was found to be capable of maintaining network connectivity whilst producing cumulative transceiver power savings in both the simulation- and test-bed-based experiments.

Chapter 4 presents the autonomous PlainTC prototype. The various design and implementation considerations are discussed and the chapter ends with a comparison amongst other WMN-applicable TC schemes.

# Part II

# Study of Topology Control in Wireless Mesh Networks

# Chapter 4

# Prototyping PlainTC, a CNN-based Topology Control Scheme

## 4.1 Introduction

The previous chapter assessed the feasibility of two (2) basic building blocks for a TC prototype for the I-WMN. First, the feasibility of programmatically adjusting the transceiver power level of a popular low-cost, resource-constrained I-WMN node was successfully established. Second, a CNN-based connectivity strategy was experimentally shown to be capable of maintaining network connectivity whilst producing cumulative transceiver power savings.

This chapter builds upon the foundational building blocks provided in Chapter 3 in order to present a TC prototype for the I-WMN. The prototyping of a TC scheme brings about additional complexities that were not required to be considered during previous theoretical and simulation-based evaluations. These complexities include:

- the constraints imposed upon the TC scheme by hardware limitations
- the integration of the TC scheme into the network protocol stack
- the effect of the node (router) firmware on the TC scheme architecture

- the need to create self-managing TC schemes to ensure autonomy and scalability

This chapter presents the design and implementation of an autonomous TC scheme, called PlainTC, that takes the above-mentioned complexities into account. PlainTC is intended for use as a TC prototype with which the behaviour and the effectiveness of CNN-based TC schemes for the I-WMN can be studied. It is envisaged that the PlainTC prototype plays a role in determining whether TC is capable of improving the energy-efficiency and performance of I-WMN deployments.

PlainTC's envisaged scope of operation is described in Section 7.1. Section 7.1 extracts the prototype scheme's design criteria from the design challenges previously described in Chapter 2 (Section 2.6). The design of the PlainTC scheme is contained in Section 4.4 and the integration of the proposed scheme into the existing protocol stack is considered in Section 4.5. PlainTC's implementation on an embedded firmware platform is described in Section 4.6. Additional benefits of the implementation approach are discussed in Section 4.7. Section 4.8 compares PlainTC to other TC schemes that can be considered for the I-WMN.

## 4.2 Operational Scope

The I-WMN may comprise nodes that use omni-directional or directional antenna. Examples of mixed antenna usage in the I-WMN can be found in Ishmael et al. (2008) and Johnson (2007). The placement of nodes employing directional antenna is usually planned to ensure that the antenna location, antenna height, antenna gain and directional orientation will result in optimised communications. This planning is essential as these directional antenna are used to facilitate long-distance communications usually between physically separate subnetworks within the I-WMN. The most common manifestation of this setup is of an I-WMN spanning multiple villages with each village connected to other villages using a long-distance, directional mesh link.

Thus, a significant degree of planning is involved with the use of nodes with directional antenna as described in Sen and Raman (2007). Little or no planning is associated with backbone nodes employing omni-directional antenna as these nodes tend to be co-located with existing infrastructure and communicate with backbone nodes in closer proximity. Site optimisation is therefore of reduced concern for backbone nodes employing omni-directional antenna. PlainTC is meant to be used on these nodes as a means of addressing the lack of communications planning or optimisation associated therewith.

The goal of the PlainTC scheme is to improve the energy-efficiency of the I-WMN by producing cumulative transceiver power output savings whilst maintaining network connectivity. This is achieved by targeting those backbone nodes that employ omni-directional antenna for transceiver power output savings. Such nodes possess a greater scope for producing savings in transceiver power output as they are not usually associated with rigorous communications optimisation and are significant sources of communications interference due to their expanded, omni-directional interference range.

PlainTC may be employed on backbone nodes that employ directional antenna but it is highly unlikely to produce transceiver power output savings since the node's transmission parameters tend to be optimised for communications with a single neighbouring (destination) node[1]. PlainTC's use of the CNN connectivity strategy is likely to increase the transceiver power output level for these nodes in the search for additional neighbours.

## 4.3 Design Criteria

This section details the first step in the transformation of the TC challenges encountered in Section 2.6 into a TC prototype. A core set of design criteria for the proposed scheme are presented, from which an algorithm is developed. Some of these design criteria have been described in Santi (2005) as *"ideal features of a TC protocol"*. They are:

[1]A common occurrence in deployed rural I-WMNs found in the literature.

- *Maintaining network connectivity*. The infrastructure WMN usage scenarios presented in Akyildiz and Wang (2005) suggest that all backbone nodes should remain available and connectivity between nodes must be ensured at all times.

- *Achieving cumulative transceiver power savings*. The use of the smallest possible transceiver power output at each I-WMN node will result in cumulative transceiver power savings. Such savings are usually equated with a corresponding extension of network lifetime if the I-WMN nodes are battery-powered. Battery-powered I-WMN nodes are not an uncommon phenomenon and such instances have been described in Farbod and Todd (2007); Coates and Delwiche (2009); Wu et al. (2011); Mannweiler et al. (2012); Johnson (2013); Rey-Moreno et al. (2013).

- *Using a fully distributed scheme*. The lack of centralised control within the I-WMN necessitates a distributed approach to TC. Each backbone node is responsible for adjusting its transceiver power output within operational constraints.

- *Using locally available information*. This information is usually collected during the normal operation of other protocols resident within the protocol stack. The use of locally available information promotes scalability and reduces the communication overhead generated by a TC scheme. It will also be shown in Chapter 7 that locally available information can be used to respond to changes in context that occur at both a network and node level.

- *Maintaining a small node degree*. A small node degree implies the use of a correspondingly small node transceiver power, which implies less interference with neighbouring nodes and the achievement of cumulative transceiver power savings. Small node degrees have been shown to be capable of maintaining network connectivity whilst producing cumulative transceiver power savings (refer to Chapter 3).

- *Supporting heterogeneous I-WMN nodes*. The I-WMN may comprise nodes with various wireless transceiver and other hardware specifications, resulting in a backbone with heterogeneous nodes. For example, some backbone nodes may possess

GPS capabilities whilst other nodes do not. Thus, TC schemes must cater for such heterogeneity in both the capabilities and specifications of potential backbone nodes. Simpler, low-complexity TC schemes will therefore lend themselves to execution on a variety of I-WMN nodes, thus aiding in the appeal of the TC scheme to both existing and future I-WMN deployments.

## 4.4 PlainTC

PlainTC is intended to be a practical, low-complexity TC scheme for the I-WMN. The scheme has the goal of improving the energy-efficiency of the I-WMN by minimising the cumulative transceiver power consumption whilst maintaining network connectivity.

PlainTC adopts the maintenance of a CNN to achieve network connectivity. This particular connectivity method has been previously shown to achieve asymptotic network connectivity in networks with uniformly distributed nodes, as the network size increases. The exploratory experiments reported in Chapter 3 showed that a CNN connectivity strategy can be employed as a building block for a TC scheme.

In summary, the CNN strategy is advantageous for the following reasons:

- it is a simple, effective method for achieving network connectivity
- it does not require the use of a GPS or additional network hardware
- it may result in cumulative transceiver power savings in the I-WMN
- it maintains the availability of all nodes in the I-WMN as sleep-cycles are not employed

PlainTC employs the CNN connectivity strategy defined in Xue and Kumar (2004) to maintain network connectivity. This work showed that the earlier convention of maintaining a constant number of neighbours would result in a disconnected network as the network size increased. The authors proved[2] the following theorem:

[2] a detailed treatment of this theorem can be found in Santi (2005)

**Theorem 1 (Xue and Kumar, 2004)** *Assume that n nodes are uniformly placed at random in* $[0,1]^2$*, and let* $G_k^+$ *be the symmetric supergraph of the k-neighbours graph built on these nodes. There exists a constant c, with* $c > 0$*, such that*

$$\lim_{n\to\infty} P(G_{c\log n} isconnected) = 1$$

where $n$ refers to the number of network nodes. The authors provided an explicit value for $c$, which is $5.1774 + \varepsilon$, where $\varepsilon$ is an arbitrarily small positive value.

Theorem 1 gives the upper bound to the CNN in networks with uniformly distributed nodes as stated in the corollary below:

**Corollary 1** *Assume that n nodes are uniformly placed at random in* $[0,1]^2$*, and let* $G_k^+$ *be the symmetric supergraph of the k-neighbours graph built on these nodes. The CNN for connectivity is*

$$k = c\log n$$

The authors proved that the number of one-hop neighbours ($k$) to be maintained needed to grow like $\Theta(log\ n)$ and that the maintenance of $k = 5.1774\ log\ n$ nearest neighbours are necessary and sufficient for connectivity w.h.p. Guaranteeing network connectivity requires the maintenance of $k = n-1$ nearest neighbours but this is a highly unrealistic requirement in ad hoc networks. Providing network connectivity w.h.p can reduce the recommended value of $k$ to $\theta(log\ n)$, thus providing an exponential saving.

These mathematical results for $k$ were obtained using an ideal environment of uniformly distributed nodes within a unit square. However, the experiments contained in Chapter 3 established the effectiveness of the maintenance of $5.1774\ log\ n$ neighbours in achieving network connectivity whilst achieving cumulative transceiver power savings in test-bed and simulation scenarios that differ from that used in Xue and Kumar (2004). Therefore, using the number of neighbours specified in Xue and Kumar (2004) provides a robust strategy for maintaining network connectivity whilst producing cumulative transceiver power savings in the I-WMN.

The use of the strategy defined in Xue and Kumar (2004) requires knowledge of the current network size. This information is easily available in both IEEE 802.11s compliant[3] and non-compliant networks.

Current outdoor, community-based WMN deployments are not compliant with the IEEE 802.11s standard[4]. However, knowledge of the current network size is easily obtained as proactive routing protocols have been found to be more effective for these deployments. Proactive routing protocols maintain routes to every possible destination in the I-WMN and thus the size of the backbone network can be derived from the number of destinations recorded in the routing table. This means that PlainTC is compatible with current standards-non-compliant I-WMN deployments that employ proactive routing protocols.

The size of the I-WMN can also be obtained in future networks that comply with the IEEE 802.11s standard. This can be achieved by using mechanisms prescribed by the standard such as forcing the registration of each backbone node with a root portal and subsequently having the portal broadcast the number of registered nodes. This approach will work with reactive, proactive and hybrid routing protocols. An alternative to the root portal broadcast solution is to employ the optional proactive RA-OLSR forwarding protocol which will cause backbone nodes to automatically maintain a forwarding table containing a path to every other backbone node. The number of destinations recorded in the forwarding table is equivalent to the size of the backbone network. Thus, PlainTC will be compatible with future standards-compliant WMN deployments as well.

PlainTC's execution logic is listed in Algorithm 1. PlainTC maintains network connectivity whilst bounding a node's degree which, according to the TC literature and the exploratory experiments in Chapter 3, will result in cumulative transceiver power savings. Thereby, the two primary objectives of TC schemes in general and PlainTC in particular, are achieved.

---

[3] see Chapter 2, Section 2.5 for a description of the IEEE 802.11s mesh networking standard

[4] The IEEE 802.11s standard is currently being implemented by a community of developers. Early releases are available at `http://open11s.org` but a complete implementation of the standard is still awaited.

**Algorithm 1** Execution Logic of PlainTC

```
no_nodes ← getRoutingTableSize()
current_no_neighbours ← getNeighbourhoodSize()
CNN ← ⌈5.1774 log(no_nodes)⌉
max_tx_power ← 78qdBm
min_tx_power ← 1qdBm
current_tx_power ← getCurrentTxPower()
tx_power_change_level ← 3qdBm
if current_tx_power ≤ max_tx_power then
    if current_no_neighbours > CNN then
        if (current_tx_power − tx_power_change_level) ≥ min_tx_power then
            current_tx_power ← current_tx_power − tx_power_change_level
        end if
    end if
    if current_no_neighbours < CNN then
        if (current_tx_power + tx_power_change_level) ≤ max_tx_power then
            current_tx_power ← current_tx_power + tx_power_change_level
        end if
    end if
end if
```

The execution logic consists of four phases, each one corresponding to the *Autonomic Control Loop* described in Dobson et al. (2006) and shown in Figure 4.1. This framework is a useful tool in the design of autonomous systems such as PlainTC. Autonomous systems must collect information to determine the current situation or context in which they operate. The collected information is analysed to inform the adaptation decisions to be taken. These decisions are subsequently implemented to complete the adaptation response.

PlainTC is designed to employ the *Autonomic Control Loop* as follows. In the first phase, PlainTC collects the current node, neighbourhood and network states. These states are composed of low-level variables such as the current transceiver power level, number of neighbouring nodes and the total backbone network size. Thus, the current context of both the node and the network can be ascertained. This contextual information is then analysed in the second phase to determine the required CNN to be maintained. The latest CNN value is then appended to the contextual information that has already been collected. PlainTC now attempts to match the current number of one-hop neighbours with the required CNN in the third phase. In the event of a mismatch,

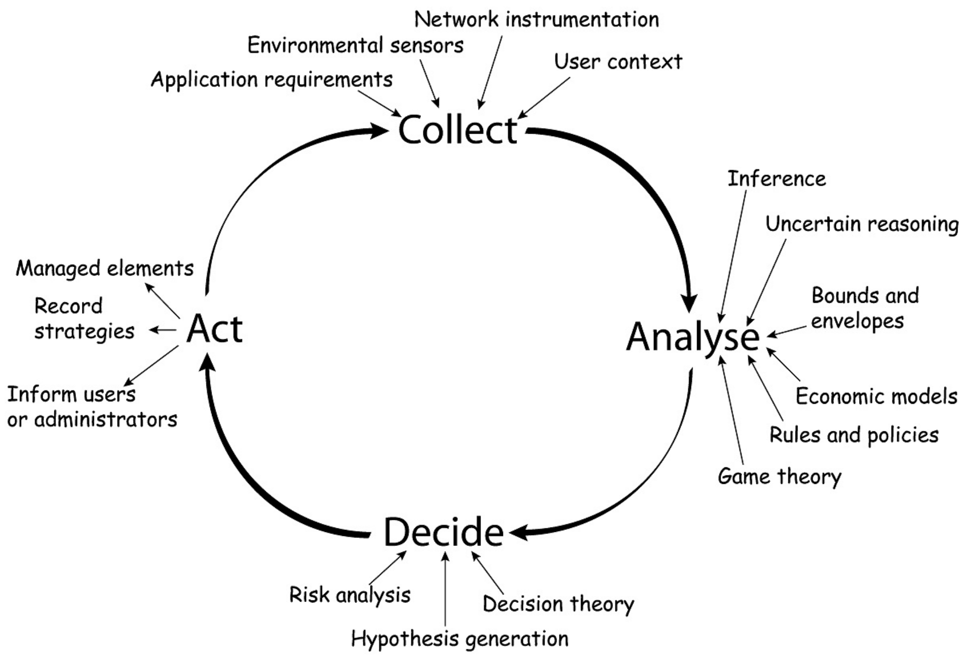



FIGURE 4.1: Phases of the Autonomic Control Loop for network scenarios (Dobson et al., 2006)

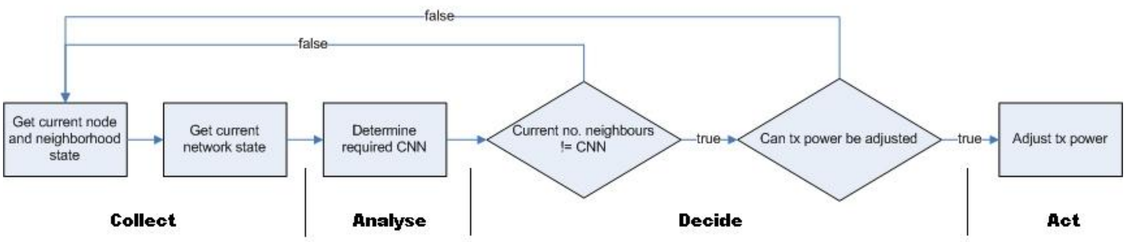



FIGURE 4.2: Flowchart depicting PlainTC's execution logic

PlainTC will ascertain whether the proposed change in transceiver power level is possible at the current power level. If a power level change is possible then the proposed change will be acted upon in the final phase and the new transceiver power level will be used. This process is meant to be executed regularly with Figure 4.2 depicting a single execution cycle.

PlainTC is distributed in nature as each backbone node can execute the scheme locally using the network size and neighbourhood size information that can be derived from information contained in the node's routing table. Support for heterogeneous nodes is provided via the minimal information required for the scheme to operate. I-WMN nodes of differing specs can execute PlainTC as long as they all employ the same proactive routing protocol. This pre-requisite is assured in deployed WMNs as a common routing protocol is necessary for the backbone to operate. Thus, a common, proactive routing protocol renders PlainTC device agnostic.

## 4.5 Integration into the Logical Protocol Stack

A traditional networking approach would suggest that a TC scheme be integrated into the protocol stack in one of two ways: (1) either as a new layer between two existing layers, or (2) as a sub-layer within an existing network layer. The first approach is advocated by Santi (2005) and used in Narayanaswamy et al. (2002) and Kawadia and Kumar (2005) whilst, to the best of our knowledge, there are no TC schemes that are resident as a sub-layer within an existing stack layer.

Both approaches fit into the conventional, horizontally-layered network protocol stack but neither is able to take advantage of the potential benefits of the cross-layered design paradigm. PlainTC is thus designed as a vertical protocol layer (see Figure 4.3) as opposed to the horizontal, layered approach advocated in Santi (2005). This choice is motivated by the flexibility afforded by a vertical design. A vertical design allows for the appearance of a conventional horizontally-layered protocol stack when the vertically-aligned layer communicates with only one horizontally-aligned layer. In addition, the vertical design supports the cross-layered design paradigm as the vertically-aligned layer is able to communicate with multiple horizontal layers and incorporate information from multiple layers to support decision-making.

PlainTC is designed to communicate with both the Physical Layer and the Network Layer. Communication with the Physical Layer is for the purpose of changing the transceiver power level of the backbone node when the need arises. Communication with the Network Layer is meant to monitor both the network size and the number of one-hop neighbours in order to trigger PlainTC's adjustment of the node's transceiver power output, if necessary.

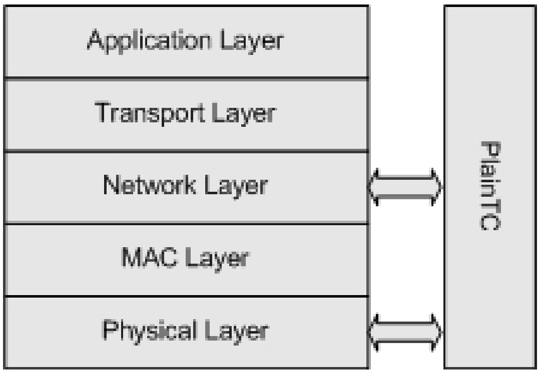



FIGURE 4.3: Placement of PlainTC with regards to the Network Protocol Stack

## 4.6 Implementation of PlainTC

Linksys WRT54GL nodes are used as the hardware platform on which to deploy PlainTC. These devices are popular backbone nodes for rural, community-based I-WMN deployments due to their low cost, rugged reputation and easily available tutorials and related support documentation.

The Linksys nodes are transformed into I-WMN nodes via the use of the OpenWRT firmware (Fainelli, 2008). This firmware is a stripped-down version of the Linux OS that caters for the limitations imposed by embedded devices and wireless routers such as the Linksys WRT54GL. Apart from common embedded Linux tools such as uClibc, Busybox and a shell interpreter, a package manager is also provided. Due to OpenWRT's modular design, the Linux kernel can be optimised to suit the underlying hardware platform whilst allowing the user-space environment to remain unaffected (requiring only a re-compilation).

Mesh networking functionality is enabled by the installation of the necessary routing and network management packages via the in-built package manager. The OpenWRT firmware also allows access to the NVRAM (non-volatile random access memory) partition of the Linksys node. This partition is the equivalent of the secondary storage facilities available on all possible I-WMN nodes. User-space packages can interact with the NVRAM partition. The 64KB NVRAM partition stores configuration variables that span the entire logical protocol stack and is thus a ready-made source of cross-layer optimisation data. It is also possible to alter the values of configuration variables.

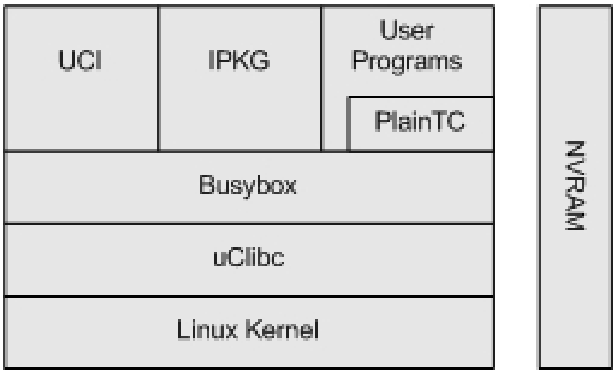



FIGURE 4.4: Implementation Architecture, adapted from (Fainelli, 2008)

As the Linksys nodes were deployed with the OpenWRT firmware to provide the mesh networking capabilities, PlainTC has to exist within the OpenWRT ecosystem depicted in Figure 4.4. PlainTC is implemented as a user-space application that is initiated at node start-up and executes at two-minutes intervals thereafter. The application is required to interact with the configuration variables (stored within the NVRAM partition) that control the magnitude of the transceiver power output of the Linksys nodes as well as with the OLSR routing protocol that is usually also implemented as a user-space application. This interaction is depicted in Figure 4.5. PlainTC relies upon the topology information collected during OLSR's normal operations. The total number of backbone nodes derived from OLSR's routing table is used to determine the appropriate CNN to be maintained. If the transceiver power output requires modification then the value of the state variable associated with the node's transceiver power level is modified. This modification forces an OpenWRT firmware trigger that sets the updated value of the state variable on the wireless transceiver hardware. The change of transceiver power level is performed without having to perform a node reboot.

No modifications to the OpenWRT firmware were necessary and therefore the advantage of the above-mentioned implementation approach was that the benefit of a cross-layer approach is gained without the need to re-engineer the protocol stack by specifying an additional protocol layer, its inter-layer interfaces and the various communication messages that would be required, thus conforming to the logical implementation architecture shown in Figure 4.3.

In summary the following implementation benefits are obtained:

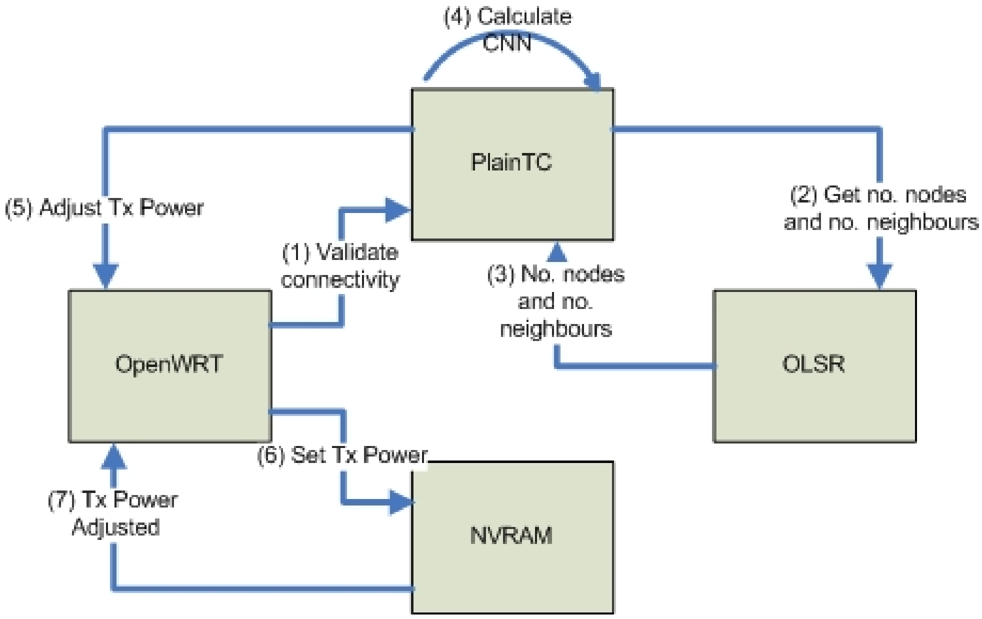



FIGURE 4.5: PlainTC's Interactions with other System Elements

(i) the de-coupling of PlainTC scheme from the traditional protocol stack layers

(ii) no new interfaces between layers being required

(iii) no new protocol messages that require defining

## 4.7 Additional Benefits of PlainTC's Implementation Approach

PlainTC has been conceived of as being a computationally simple, standards-agnostic TC scheme that results in little re-engineering of the protocol stack and node firmware. PlainTC's implementation approach does possess some additional advantages when compared to existing WMN-applicable TC prototypes found in the literature. These additional benefits are:

- *No data collection at every supported transceiver power levels*. The TC prototypes described in Sethu and Gerety (2009) and Kowalik et al. (2008a) require that routing information be collected for every supported transceiver power level.

This information is added to routing table entries to facilitate routing at the lowest possible transceiver power levels. The schemes devised in Kawadia and Kumar (2005) go even further by maintaining routing tables at each supported transceiver power level.

- *No additional communication requirements*. The data collection process described above involves the broadcasting of beacon messages. These beacon messages constitute the communication overhead produced by these TC schemes which consumes precious bandwidth, thus affecting QoS. It must also be noted that the communication overhead produced by the schemes in Sethu and Gerety (2009); Kowalik et al. (2008a); Kawadia and Kumar (2005) is additionally magnified by the broadcasting of beacon messages at each supported transceiver power level.

- *No extension of routing tables*. Existing TC schemes (Sethu and Gerety, 2009; Kowalik et al., 2008a) modify the routing tables that are created by routing protocols. The lowest transceiver power level to reach either a destination or an intermediary of the destination is stored in the routing table. This transceiver power level information is not natively part of routing tables and thus the storage of this information requires modifications to the routing table.

  The work in Kawadia (2004) extends the routing table to support on-demand and deferred route entries. On-demand entries are for routes where packet queueing is supported in the event that a route is currently unavailable. Deferred entries are for routes that are yet to be discovered.

## 4.8 Comparing PlainTC to other TC prototypes that can be considered for an I-WMN

A small number of TC prototypes applicable to the I-WMN can be found in the literature. The applicability of these prototypes is determined by the absence of sleep-based mechanisms, the potential to produce cumulative transceiver power savings, independent or autonomous operation and the suitability of the hardware platforms on which the original evaluations, described in the literature, occurred. These earlier prototypes have been previously discussed in Chapter 2 (Section 2.4). This comparison, however, focuses upon implementation issues. A summary of the key implementation characteristics of these prototypes can be found in Table 4.1.

The COMPOW, CLUSTERPOW and MINPOW schemes are based on the maintenance of multiple routing tables: one for every power level supported by the wireless hardware (Kawadia, 2004). PlainTC, however, requires the maintenance of only one routing table since broadcasts at multiple power levels pose synchronisation challenges. The maintenance of up-to-date routing tables for every power level requires the almost constant switching of power levels in order to send probe packets or HELLO messages at each power level. The ability of COTS WiFi hardware to support the fast switching of power levels whilst simultaneously sending and receiving data traffic is in doubt as the developers of these three schemes have reported repeated hardware crashes. In addition, the appearance of discrete power levels is more prevalent amongst wireless cards attached to laptops than to actual I-WMN hardware which are standalone nodes. I-WMN compatible hardware has been reported not to possess discrete power levels but are rather able to accommodate various granularities of power change.

The ConTPC prototype (Kowalik et al., 2008a) also sends probe packets or HELLO messages at every supported power level. However, ConTPC uses these probes to determine the effect of the various power levels on the data delivery rate for each incoming link. Thus, per-link transceiver power adjustment is practiced, which increases the complexity of the scheme. In addition, a stable transceiver power level is difficult to maintain since this level will vary for each outgoing link, eventually resulting in an unstable network topology and therefore increased routing updates. PlainTC employs per-node transceiver power adjustment where the same power level is used for every outgoing link. This results in a more stable network topology with fewer routing updates in response to transceiver power changes.

Experimental results obtained with ConTPC showed that reductions in power levels adversely affected link quality and the data delivery rate. ConTPC is designed to avoid reductions in link quality and thus the designers of the prototype report that no power savings were achieved. PlainTC contains a power saving bias as it is targeted at battery-powered I-WMNs. In addition, PlainTC is designed to promote a stable network topology by avoiding per-link power adjustment.

A study reported in Valera et al. (2008) uses a commercially-available wireless router platform to test the effect that transceiver power adjustment has on both Transport-Layer (UDP) packet performance and Network-Layer (routing protocol) performance. A common power level is manually calculated and the transceiver powers of the nodes are manually adjusted. PlainTC differs from this approach by automating the power level calculations and subsequent transceiver power adjustment. In addition, PlainTC does not use a common power level for all network nodes as this approach is susceptible to the presence of outlying nodes that force a higher-than-necessary power level on all other devices. The use of a common power level does however mitigate against the hidden-node problem.

PlainTC thus differs in three (3) ways from these existing prototypes. Firstly, the maintenance of multiple routing tables is avoided. Secondly, the number of changes to the transceiver power level of a backbone node is reduced by adopting a per-node approach to the power level employed. Lastly, PlainTC prioritises the production of cumulative transceiver power output savings in addition to maintaining network connectivity.

TABLE 4.1: Comparison of Prototyped TC Schemes found in the literature

| | **PlainTC** | **COMPOW** | **CLUSTERPOW** | **MINPOW** | (Valera et al., 2008) | **ConTPC** |
|---|---|---|---|---|---|---|
| **Objective** | maintain connectivity whilst minimising transceiver power consumption | utilise lowest common power level such that connectivity is maintained | packets routed to destinations using non-increasing power levels | globally optimises total energy consumption | maintain connectivity whilst minimising transceiver power consumption | to reduce transceiver power only if a link's delivery rate is not adversely affected |
| **Connectivity Mechanism** | maintenance of a CNN | homogeneous transceiver power assignment | ensures destination can be reached using non-increasing power levels at each hop | ensures the use of minimum power levels such that network connectivity is maintained | manual calculation of the CTR | maintenance of original neighbours |
| **Transceiver Power Savings** | yes, if CNN maintenance does not require maximum power at every backbone node | yes, only if the common power level is not the max. power level | yes, if the use of non-increasing power levels actually results in lower power levels being used | yes, if a node does not utilise the max. power level | yes, if CTR is less than the max. power level | yes, if delivery rates at max. power can be sustained using a lower power level |
| **Messaging Requirements** | none, uses existing routing protocol messages | multiplies existing routing protocol messages by the number of power levels. Also adds communication between each routing protocol instance and the COMPOW agent | magnifies existing routing protocol messages by the number of power levels. Also adds communication between each routing protocol instance and the CLUSTERPOW agent | magnifies existing routing protocol messages by the number of power levels. Also adds communication between each routing protocol instance and the MINPOW agent | none since transceiver power is manually adjusted | probe messages sent at all supported power levels |
| **Cross-Layered** | yes | no | no | no | no | yes |
| **Implementation Approach** | a user-space application, loosely-coupled with underlying firmware. Used with proactive routing protocols | a user-space application with proactive routing protocol instances at each power level. An agent examines each routing table at every power level to determine optimal common power level | a user-space application with proactive routing protocol instances at each power level. An agent examines each routing table at every power level at every intermediate node to determine lowest power level to reach destination | a user-space application with proactive routing protocol instances at each power level. An agent examines each routing table at every power level to ensure minimum power level usage | manual adjustment of node transceiver powers | ConTPC implemented as a Click (Kohler et al., 2000) module. Probe packets are sent at all power levels to determine delivery rates for each incoming link at each power level. |
| **Evaluation Platform** | Linksys WRT54GL nodes, used in I-WMN deployments | Cisco Aironet 350 series wireless cards. Not I-WMN nodes | Cisco Aironet 350 series wireless cards, not I-WMN nodes | Cisco Aironet 350 series wireless cards. Not I-WMN nodes | Compex WP54G router, can be used in I-WMN deployments | Soekris net4521-based devices, can be used in I-WMN deployments |
| **IEEE 802.11s Compatible** | yes | no, modifications made to the routing table | no, modifications made to the routing table | no, modifications made to the routing table | no, CTR calculations are typically centralised | yes |

## 4.9 Chapter Summary

This Chapter has described the design and implementation decisions taken to produce the PlainTC TC prototype. The prototype was designed for battery-powered I-WMNs that are typically deployed in rural areas. These backbones tend to use directional and omni-directional antenna to fulfil differing roles. PlainTC was thus proposed for backbone nodes that employ omni-directional antenna because of the lack of planning and optimisation associated with these nodes.

PlainTC has adopted a cross-layer approach by communicating with both the Network Layer and the Physical Layer to ensure that network connectivity was maintained whilst achieving cumulative transceiver power savings. This approach allows PlainTC to be utilised both on current deployments that are not compliant with the IEEE 802.11s standard in addition to future 802.11s-compliant deployments.

The resultant prototype has adopted the popular Linksys WRT54GL WMN node and the OpenWRT firmware as the implementation platform. PlainTC is initiated at node startup and operates thereafter at regular intervals. The result is the low-complexity, standards-compatible and autonomous PlainTC prototype that has been shown to possess several benefits over previous TC prototypes. The PlainTC prototype will serve as the basis for evaluating whether improvements to the energy-efficiency and performance of the I-WMN can be achieved.

The next Chapter (Chapter 5) describes the evaluation setup and the measurement methodologies for the various experiments that are conducted as part of the afore-mentioned evaluation. Details for both the test-bed and simulation evaluations are provided.

# Chapter 5

# Evaluation Setup and Measurement Methodology

## 5.1 Introduction

The previous chapter (Chapter 4) presented the design and implementation of an autonomous, standards-compatible CNN-based PlainTC TC prototype. The prototype implemented on the Linksys WRT54GL hardware platform is a tool for studying the viability of PlainTC as an energy efficiency and performance improvement mechanism for an I-WMN.

The prototype implementation is meant to allow the study of PlainTC's run-time behaviour and its effect on the energy-efficiency and performance of the network. The lack of scalability afforded by the indoor test-bed being used necessitated the use of a simulator as a secondary evaluation tool.

The NS-2 simulator is commonly used in WMN evaluations but does not natively provide TC functionality for ad-hoc networks as the tool is architected under the assumption that a node's transceiver power output does not change during the simulation. Significant changes must be made to NS-2's codebase when introducing a TC scheme and these changes are usually incompatible with subsequent NS-2 releases.

An alternative approach is to adopt a simulation tool that is specifically designed for TC research. This study employs the Atarraya simulator described in Labrador and Wightman (2009) to create connected I-WMNs and compute the transceiver power outputs required to maintain the CNNs specified by PlainTC. Atarraya's traffic generation and data analysis capabilities are not as mature as NS-2's and therefore a unique two-stage simulation process was devised. Thus, the strengths of both Atarraya and NS-2 were accommodated to simulate the network topologies created by PlainTC. Due to loose coupling, subsequent improvements to either tool does not affect the functioning of this two-stage simulator.

This chapter details the setup of both the test-bed and simulator as evaluation platforms. The test-bed setup is described in Section 5.2 and the simulation setup is described in Section 5.3. The methodology used to measure the various performance metrics are detailed in Section 5.4. This section provides the measurement methodologies for both the test-bed and simulator platforms.

## 5.2 Test-bed Setup

The indoor mesh test-bed operated in a ground-floor laboratory in a 3-storey office building at the University of Zululand's main campus. The test-bed network operated in 802.11g mode on channel 6 in order to mitigate against interference caused by a separate WLAN that was operational within the building.

The mesh test-bed consisted of 14 nodes placed in a $6mx4m$ area as shown in Figure 5.1. The node placement was determined by the availability of plug points, which is analogous to the coupling of nodes with existing infrastructure in real-world deployments. Each node in the mesh backbone consisted of a mains-powered, Linksys WRT54GL router with the OpenWRT-based Freifunk firmware (version 1.7.2) used to provide mesh networking functionality. The Freifunk firmware uses the OLSR routing protocol by default as this protocol has been successfully employed in large-scale WMN deployments.

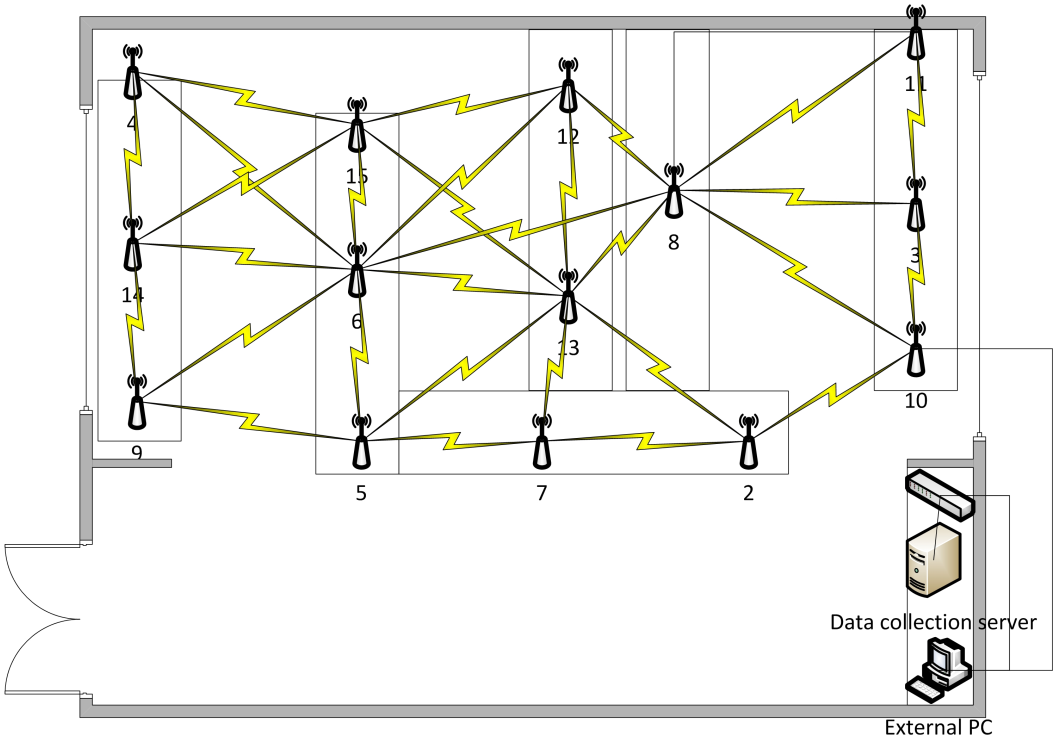
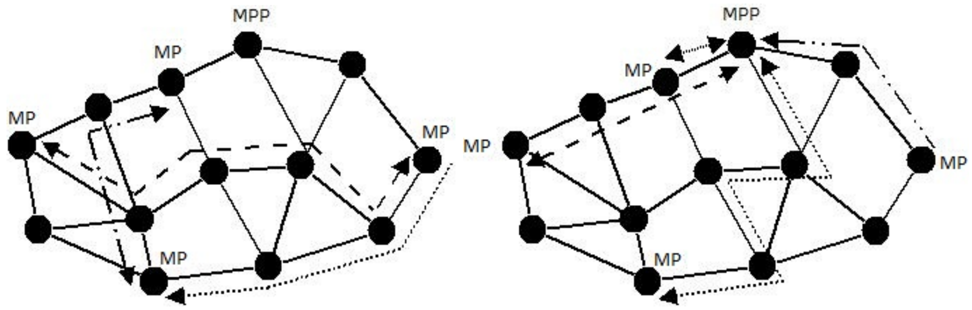



FIGURE 5.1: Testbed Layout

The Linksys WRT54GL routers possessed a 200MHz processor, 16Mb of RAM, 4Mb flash memory and a Broadcom 802.11b/g radio chipset. The wireless chipset allowed transceiver power output levels to be set from 1dBm to 19.5dBm. The latter value is the maximum power output recommended by the manufacturer.

Each node was connected via Ethernet through a switch to a central server. The use of the Ethernet port for data collection was to allow for network traffic to flow over the wireless mesh network and for performance data and other statistical information to flow over the Ethernet network. Thus, the performance data did not interfere with the data flowing over the wireless mesh network. The central server also synchronised the clocks of all the network nodes in order for the collected data to be accurately time-stamped.

Custom data collection scripts were written and installed within each test-bed node. These scripts were designed to amalgamate the time-stamped data from the node and the routing protocol at one-second intervals. This data was "pushed" by each test-bed node to the central data collection server where it was stored whilst awaiting further analysis. The data collected from the individual nodes included their current transceiver




(a) Intra-mesh communication (b) Portal-oriented communication

FIGURE 5.2: Communication Scenarios

power output, network size, neighbourhood size and the link quality associated with their various neighbouring nodes.

Data traffic was generated by using the *iperf* traffic generator installed on each test-bed node. The central server remotely controlled the operation of these traffic generators. The testbed supported two traffic scenarios: the Intra-mesh traffic scenario where the source and destination are within the test-bed backbone and the Portal-oriented traffic scenario where either the source or destination was resident outside of the test-bed backbone. For the Portal-oriented scenario, a PC accessible via the designated Gateway node (node 10 in Figure 5.1) was setup to act as an external source or destination. These two scenarios are depicted in Figure 5.2.

The *iperf* tool allowed for the generation of both UDP- and TCP-based traffic. The UPD-based traffic was used in the intra-mesh scenario to emulate activities such as LAN gaming. The TCP-based traffic was meant to emulate Web activities where at least 80% of all traffic is carried via TCP. The server was setup to initiate traffic flow from 90% of the network nodes to randomly-selected, destination nodes for the Intra-mesh scenario. The server initiated traffic flows from 90% of the network nodes to the external PC for the Portal-oriented scenario.

The test-bed allowed for the evaluation of both constant-size and incrementally-increasing network backbones. The constant-size networks were initiated with all the network nodes being switched on simultaneously and a ten-minute stabilisation period was allowed to pass before data was collected. The duration of the data collection in the constant-size experiments was 60 minutes. Network sizes ranged from 8-14 nodes.

The afore-mentioned experiments were designed to examine the dynamic or run-time behaviour of PlainTC when nodes were added to an existing backbone network. These experiments began with an initial network size of eight (8) nodes and edge nodes were incrementally introduced to the backbone network at ten-minute intervals. The initial 8-node network was given a ten-minute stabilisation period before data collection begins. Data collection was stopped after a duration of 70 minutes with data being collected whilst each newly-introduced node adjusted its transceiver power output. Thus, the collected data showed the response of the newly-introduced nodes to the existing network and showed the reaction of the existing nodes to the introduction of a new node. With data being collected at one-second intervals, a total of 600 data points were collected by each node for each observed metric for each evaluation run. Each experiment was repeated five (5) times as the volume of data collected posed a storage challenge.

## 5.3 Simulation Setup

The test-bed is primarily used to observe PlainTC's dynamic, run-time behaviour and provides a new perspective on evaluating the performance of a TC scheme. The simulator is primarily employed to evaluate PlainTC from the traditional perspective where the well-known performance metrics and metric measurement methodologies found in the TC literature are employed.

The NS-2 simulation tool is a commonly used wireless ad-hoc network simulator. The tool offers extensive support for traffic models, node models and energy models. However, NS-2 does not easily support changes to a node's transceiver power output level during run-time. Thus, TC schemes cannot be directly simulated using this simulation tool.

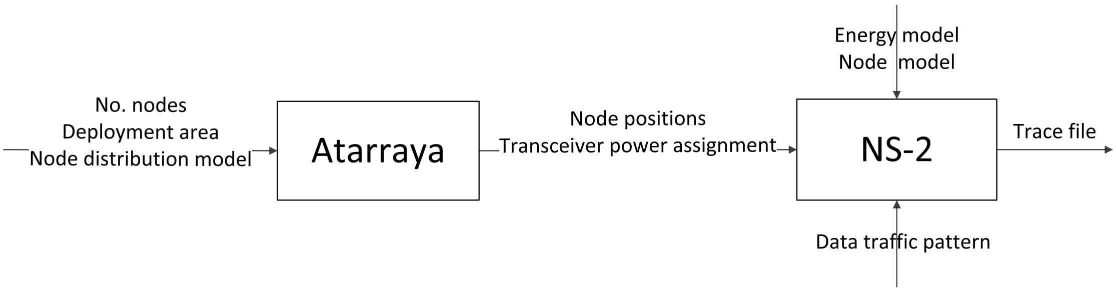



FIGURE 5.3: Two-phase Simulation process

The Atarraya simulation tool (Labrador and Wightman, 2009) is designed as a simulator for TC schemes for Wireless Sensor Networks. This tool provides support for changes to transceiver power output levels during runtime but its data traffic and node model support is lacking.

A two-stage simulation process was designed to accommodate the strengths of both the Atarraya and NS-2 simulation tools. This process is depicted in Figure 5.3. The first stage used the Atarraya simulation tool. Atarraya was modified to support an I-WMN and the PlainTC scheme.

Atarraya was used to generate the network topology from three (3) primary inputs: the number of nodes in the backbone network, the deployment area and the node distribution model. The number of backbone nodes varied from 20 to 120 whilst the deployment area and the node distribution model were fixed. Nodes were uniformly distributed within a $1000mx1000m$ deployment area. Figure 5.4 depicts the network topology that Atarraya generated from these inputs. The Atarraya topology generation process also ensured the creation of a connected network topology.

The generated network topology allowed for the selection of the TC scheme to be employed on the backbone network topology. This interface is depicted in Figure 5.5. Both the Max. Power and PlainTC schemes were available for selection[1]. The selected TC scheme was executed and the necessary transceiver power levels were assigned to every backbone node. Figure 5.6 depicts the resultant network topology if the PlainTC option is selected[2]. The first stage of the simulation setup was thus completed.

[1]The Max. Power scheme is equivalent to the “No TC Protocol” option.
[2]This figure can be compared with Figure 5.4 which depicts the original network topology

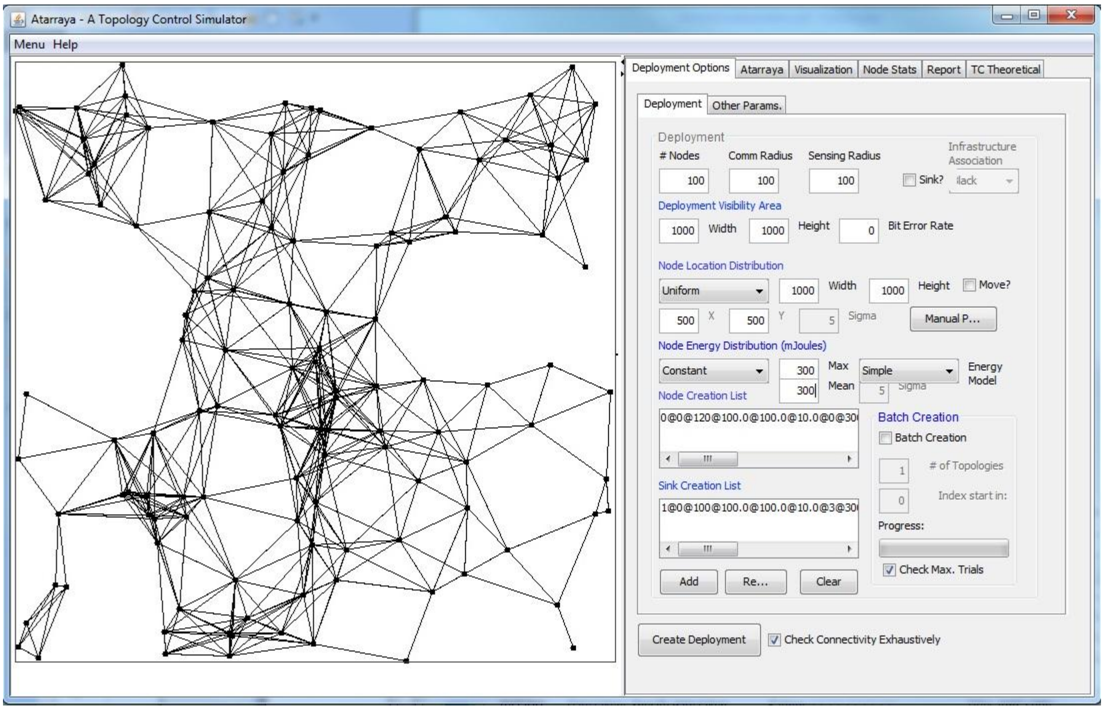



FIGURE 5.4: Network topology generated by Atarraya

The node positions obtained from Atarraya were imported into NS-2 scenario files containing the X- and Y-coordinates for all the backbone nodes. The transceiver power output levels associated with each node were imported into an NS-2 script file. The script file was supplemented with the energy consumption model, the node model and the data traffic pattern. This represented the start of the second stage of the simulation setup.

The power consumption and node models were informed by the specifications of the Linksys WRT54GL node. The maximum transmission range was however limited to 100$m$ to prevent the need for large simulated deployment areas.

The data traffic model for NS-2 simulations is commonly created using the accompanying *cbrgen.tcl* traffic generator. This traffic generator support both UDP- and TCP-based traffic but only within the Intra-mesh scenario (see Figure 5.2). Support for the Portal-oriented scenario was added to the *cbrgen.tcl* traffic generator. The traffic generator was configured to ensure that 90% of nodes initiate traffic flows to random destinations in the intra-mesh scenario or to the designated mesh portal (Gateway node) in the Portal-oriented scenario.

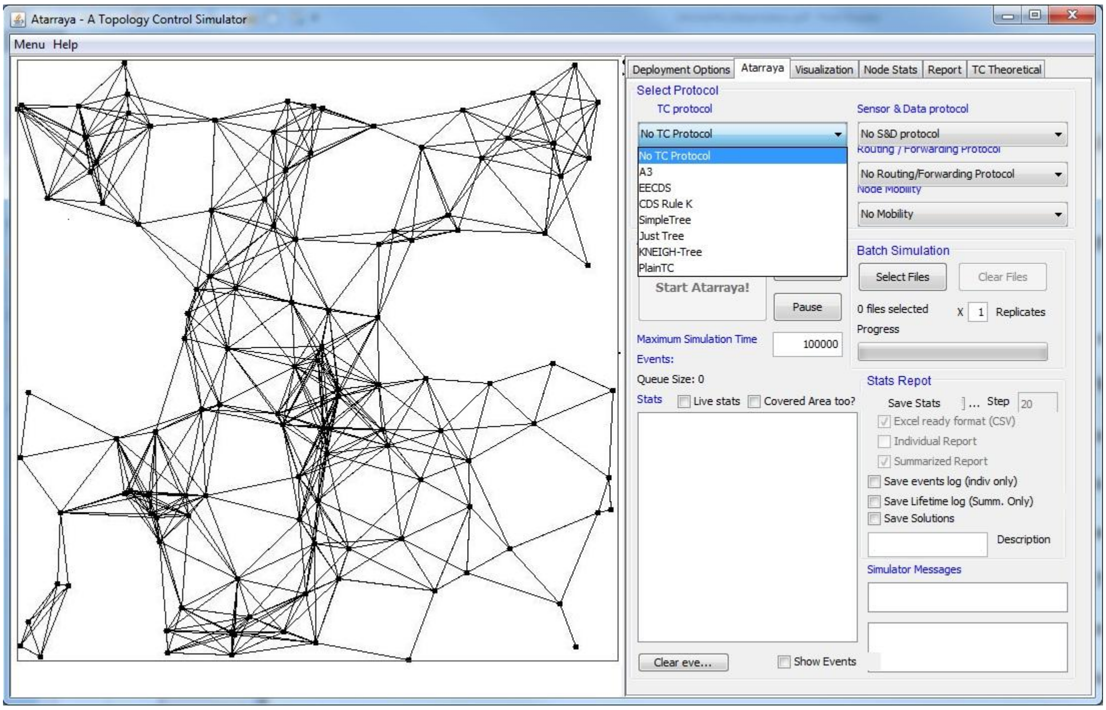



FIGURE 5.5: Selecting the TC scheme to be used

TABLE 5.1: Simulation Details

| | |
|---|---|
| Simulation Time | 120 seconds |
| Network Size | 20–120 nodes |
| Network Area | 1000m x 1000m |
| Routing Protocol | OLSR |
| Traffic type | UDP/TCP with 90% of nodes as traffic sources |
| Traffic rate | 40 packets per second with a max. of 1000 packets |
| Max. transceiver range | 100m |
| Initial energy | 1.0 J |
| Transmit Energy | 0.6W |
| Receive Energy | 0.3W |

Each traffic flow was configured to run for the time taken to deliver 1000 data packets. Packets were generated by the source node at a rate of 40/sec with each packet containing a 512 byte payload. A summary of the simulation details can be found in Table 5.1.

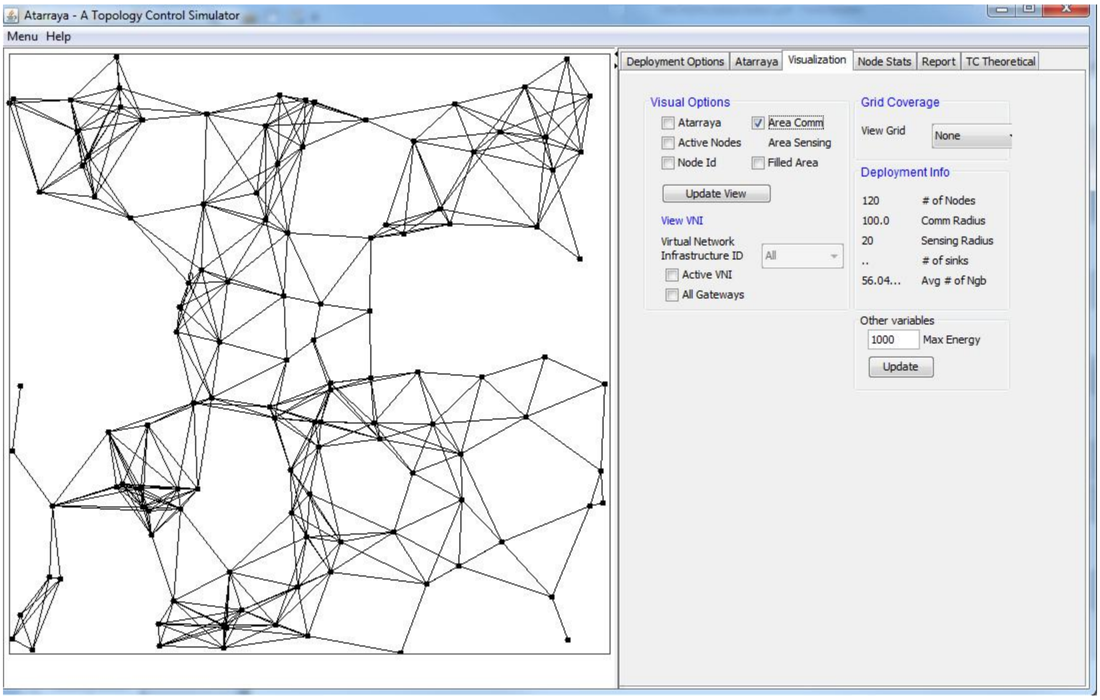



FIGURE 5.6: Network topology created by PlainTC on the Atarraya simulator

## 5.4 Measurement Methodology

This section contains descriptions of how the various metrics were measured. The procedures used to measure many of these metrics were highly similar on the test-bed and simulator. In cases where the test-bed and simulation measurement methodology for a particular metric differ, both measurement methodologies are presented.

### 5.4.1 Network Connectivity

Network connectivity is defined as the number of source-destination pairs that exist within the backbone network. The backbone network is considered fully-connected if every node has a route to every other node in the backbone. Thus, a maximum of $n^2 - n$ source-destination pairs can exist where $n$ represents the number of nodes in the backbone network.

The proactive OLSR routing protocol is designed to store the best route for each possible destination node. These routes are stored in a routing table where the number of entries in the table corresponds to the number of backbone nodes that can be reached. Each node was programmed to report the number of entries in their routing table. The sum of the number of entries for all network nodes represented the total numbers of available source-destination pairs for that experiment iteration. The average number of available source-destination pairs across all experiment iterations was recorded.

This procedure was used for both the test-bed and the simulation evaluation platforms.

### 5.4.2 Cumulative Transceiver Power Output Savings

The cumulative transceiver power savings represent the difference between the cumulative maximum transceiver power output levels and the cumulative transceiver power output levels assigned by PlainTC. This is alternatively represented as

$$CumulativeTransceiverPowerSavings = \sum_{i=1}^{n} Tx_{MaxPower} - \sum_{i=1}^{n} Tx_{PlainTC}(i) \tag{5.1}$$

where $n$ signifies the number of nodes in the backbone network.

The savings are also contextualised by expressing them as a percentage of the cumulative maximum transceiver power output. This can be represented as

$$CumulativePowerSavingsPercentage = 100 - \left( \frac{\sum_{i=1}^{n} Tx_{PlainTC}(i)}{\sum_{i=1}^{n} Tx_{MaxPower}} \times 100 \right) \tag{5.2}$$

Each node recorded its transceiver power output level. The sum of these power levels was noted for each experiment iteration and the average cumulative transceiver power saving was recorded. This process was repeated across the various network sizes.

The test-bed evaluation required an analysis of the time-stamped data sent by each network node. The time-stamp allows the cumulative transceiver power output level for the entire network at a particular moment in time to be calculated. The simulation evaluation requires an analysis of the node transceiver power output levels assigned by the Atarraya tool. The output provided by Atarraya provided a listing of the transceiver power output level assigned to every network node. This cumulative transceiver power output level is recorded for every experiment iteration and the process is repeated for every network size.

### 5.4.3 Average Path Length

The path length measures the change that PlainTC's reduction in node transceiver power output levels has on the length of routing paths. The path length is determined by the ETX routing metric employed by OLSR. OLSR produces routing table entries for every possible destination node. Each entry contains the length of the path to that particular destination. The average length of all the paths in all the nodes' routing tables were noted. The average across all experiment iterations was recorded and the process was repeated for every network size.

This procedure was used for both the test-bed and the simulator evaluation platforms.

### 5.4.4 Physical Layer Interference

This measurement was only undertaken on the test-bed as the NS-2 physical model is not sophisticated enough to capture the variations in Physical Layer conditions. The test-bed nodes report their recorded RSSI and noise levels at one-second intervals. The average RSSI and noise levels are recorded over the 60-minute duration of each experiment iteration. The average RSSI and noise levels across all iterations is then calculated. The procedure is repeated for each network size being considered.

### 5.4.5 MAC Layer Interference

This metric measures contention for the shared wireless communication medium. The metric captures the number of neighbouring nodes that are prevented from communicating whilst the currently-communicating node has access to the medium.

OLSR's routing table records the number of destinations that can be reached in one hop. These one-hop destinations represent immediate neighbouring nodes. The number of such nodes is reported by each node. The average size of the one-hop neighbourhood is computed across all the network nodes in each experiment iteration. The average one-hop neighbourhood size across all iterations is recorded. The recorded value represents the MAC-level interference. The procedure was repeated for the various network sizes.

This procedure was used for both the test-bed and the simulator evaluation platforms.

### 5.4.6 Spatial Re-use

Spatial re-use measures the effect of reduced transceiver power output levels on the number of concurrent traffic flows that can be supported by the backbone network. The data collection server initiates random data traffic flows at fixed ten-minute intervals. New traffic flows were initiated until the next traffic flow could not be initiated due to network saturation. The number of successfully initiated traffic flows was averaged across the experiment iterations. This procedure was repeated for all the network sizes that were considered.

A similar procedure was followed in NS-2. Traffic flows were initiated at fixed intervals run until the simulation ended. These traffic flows were created using the *cbrgen.tcl* traffic generator and stored in a traffic file. A custom analysis script inspected the simulation trace file for sequences of *IFQ* error messages associated with a single traffic flow. These messages are symptomatic of network saturation.

The time associated with the earliest recorded *IFQ* error event was noted. This time value was compared with the starting times of the traffic flows contained in the traffic file. The number of successfully initiated traffic flows was determined as the number of traffic flows initiated before the first *IFQ* error event. This number was averaged across all the experiment's iterations. The process was repeated for the various network sizes.

### 5.4.7 Packet Delivery Ratio

The Packet Delivery Ratio (PDR) is measured for the period until network saturation is achieved. This is done to remove the negative influence of the saturation effect on the result. Thus, the experiment is concerned with the pre-saturation PDR achieved by the network.

The PDR was calculated as the number of successfully delivered data packets for the network divided by the number of successful initiated packets. This value was averaged across all the experiment's iterations. The process was repeated for the various network sizes.

The overall procedure was used for both the test-bed and the simulator evaluation platforms.

### 5.4.8 Throughput

The throughput is calculated for the period until network saturation is achieved. This is done to remove the negative influence of the saturation effect on the result. Thus, the experiment is concerned with the pre-saturation throughput achieved by the network.

This throughput metric measures the cumulative rate at which packets are delivered to their intended destinations. The test-bed used the data collection server to record the throughput values for each initiated traffic flow. These values were summed for each experiment iteration and the reported value is the average of the summed network throughput values. This process was repeated for the various network sizes.

NS-2 requires an analysis of the trace file to determine the network throughput. A throughput analysis script was employed. This script extracted the number of delivered packets and the duration over which those packets were delivered. The number of delivered packets is multiplied by the size of each packet to determine the quantity of transmitted data. This product is divided by the recorded duration of the flow to produce the throughput for a single experiment iteration. The average of the throughputs across all iterations was recorded as the average throughput for that particular network size. This process was repeated for the various network sizes.

### 5.4.9 Network Lifetime

This metric measures the elapsed time until the first backbone node depletes its energy source. The definition ensures that the elapsed time represents the period for which all the backbone nodes are available.

This metric was only measured on the simulator platform as the test-bed was not battery-powered. Each simulated node began with a $1J$ energy supply and this supply was gradually depleted during the operation of the node. The NS-2 trace file recorded the energy values for each network node. A custom analysis script was used to scan the trace file for instances of nodes associated with a $0.00J$ energy value. The elapsed time until the first such instance is found is recorded as the network lifetime. This process was repeated for all network sizes.

### 5.4.10 Energy Cost of Data Transfer

This metric measures the amount of energy expended to transmit a KB of data. Thus, the energy efficiency of data transmission can be determined. The energy cost of data transfer is defined as the total energy available to the backbone network divided by the average amount of data transmitted over the network's lifetime[3]. An alternative representation can be found below

$$AverageEnergyCost/KB = \frac{TotalAvailableEnergy}{AverageThroughput \times NetworkLifetime} \tag{5.3}$$

The $TotalAvailableEnergy$ value is the sum of the energy supplies of all the network nodes. The $AverageThroughput$ value was calculated as described earlier in Section 5.4.8 whilst the $NetworkLifetime$ value was calculated as described in Section 5.4.9.

[3]The network lifetime is chosen over the simulation time in order for the average energy efficiency of the full network backbone to be determined.

This measurement was only performed on the simulator as the test-bed was not battery-powered so no $NetworkLifetime$ value can be ascribed. A separate study that investigated the energy cost associated with battery-powered nodes can be found in Oki (2013). The study employed the same indoor test-bed network described in this chapter.

## 5.5 Chapter Summary

This chapter described the setup of both an indoor test-bed and a two-phase simulator platform. Both the Atarraya and NS-2 simulators are used with the output of the Atarya TC simulator being used as input into the NS-2 simulator. The Max. Power and PlainTC schemes will be compared using a set of nine (9) evaluation metrics. The associated experiments and the details of the measurement methodologies employed for each metric were described.

The next chapter (Chapter 6) uses the PlainTC prototype and simulation implementations as the basis for experimentation. The chapter aims to assess the viability of PlainTC as a means for improving the energy-efficiency and performance of deployed I-WMNs. Chapter 6 will take a look at the dynamic behaviour of the PlainTC prototype. This represents a hitherto unique view of the dynamic, run-time execution of a TC prototype. In addition, PlainTC's performance against well-known TC metrics will be presented in the next chapter.

# Chapter 6

# Assessing the Viability of PlainTC

## 6.1 Introduction

The previous chapter (Chapter 5) detailed the evaluation setup employed in this study. In addition, the measurement methodologies employed for the various evaluation metrics were described.

This chapter presents the results obtained from both the test-bed and simulation evaluations. The experiments reported in this chapter assess PlainTC's energy-saving and data-carrying capabilities to ultimately determine whether PlainTC is a viable mechanism for deployed I-WMNs. These twin capabilities are generally associated with TC schemes in the literature and are explicitly stated in the definition of TC found in Santi (2005). Energy savings are often assumed due to the reduction in node transceiver power levels whilst the data carrying capability of the network is often shown to have improved, as found in Ramanathan and Rosales-Hain (2000); Huang et al. (2002); Jia et al. (2004); Tang et al. (2005); Li et al. (2005a,b); Hu et al. (2006); Siripongwutikorn and Thipakorn (2008); Aron et al. (2008c); Jang and Fang (2008). Thus, PlainTC is also measured against the twin objectives of achieving energy savings whilst simultaneously improving the data-carrying capacity of the I-WMN.

Two types of evaluations are to be found in this chapter. Both types of experiments are complimentary in nature and help to provide a more well-rounded assessment of PlainTC's viability than any single experiment type is able to achieve on its own.

The first type takes advantage of the test-bed evaluation platform to observe the dynamic, run-time behaviour of the PlainTC prototype. These observations are presented in Section 6.2. To the best of our knowledge, this represents the first time that the dynamic, run-time behaviour of a TC prototype applicable to an I-WMN has been documented. The observations presented in Section 6.2 represent a contribution to knowledge as there is a lack of such observations in the literature.

The second type of evaluation resembles the traditional static performance snapshots that are found in the literature. These experiments are presented in Section 6.3. This evaluation distinguishes itself from existing evaluations in two (2) ways. The evaluation first establishes that cumulative transceiver power savings are achievable and then further reveals that a node's location within the network is correlated with the quantum of transceiver power savings that the node achieves. The majority of nodes eventually employ a transceiver power between 71% and 99% of the maximum transceiver power value. These are unique insights into the nature of the cumulative power savings achieved. The evaluation also distinguishes itself from existing knowledge by investigating the node failure profiles of the I-WMN nodes. Groups of dispersed nodes are shown to share similar lifetimes. This finding indicates the presence of routing highways and the presence of primary, secondary and, in some instances, tertiary highways.

## 6.2 Observations of the Dynamic Run-time Behaviour of the PlainTC Prototype on the Test-bed

The test-bed evaluation platform allows for the examination of the dynamic run-time behaviour of PlainTC. This represents a unique view of the execution of a TC scheme. The dynamic behaviour of PlainTC is important as it is an indicator of the scalability of this scheme. Scalability is an important characteristic of deployed I-WMNs and the scalability of a TC scheme must be determined if the scheme is to be considered for use in such a setting. In addition, the reaction of PlainTC to network events can be tracked. The lack of such observations in the literature allows the observations contained in this section to establish benchmarks that can be used for future comparative studies of the dynamic behaviours of prototyped TC schemes.

This section presents the response of PlainTC to a growing I-WMN. This is an often-encountered scenario in deployed I-WMNs as these networks are often deployed in phases and are accommodating of network expansion. The growing test-bed network allows for the observation of how PlainTC:

- consumes computational resources
- re-establishes network connectivity
- maintains the required neighbourhood size
- produces cumulative transceiver power output savings
- affects the workings of the routing protocol on individual backbone nodes

The test-bed network is initially composed of 8 nodes and a new node joins the network every ten minutes. The experiments reported below thus examine how PlainTC's iterative transceiver power adjustment process affects the metrics being measured. Additional setup details and measurement methodologies for this set of experiments have already been described in Chapter 5.

### 6.2.1 Resource Consumption

PlainTC was observed to consume 368KB of memory (2.3% of total memory) and 0.3% of the CPU contained within the test-bed nodes. Due to the localised nature of the scheme, no discernible differences in memory consumption and CPU load were observed as the network size grew. Thus, PlainTC does not negatively affect the performance of the Linksys WRT54GL nodes employed.

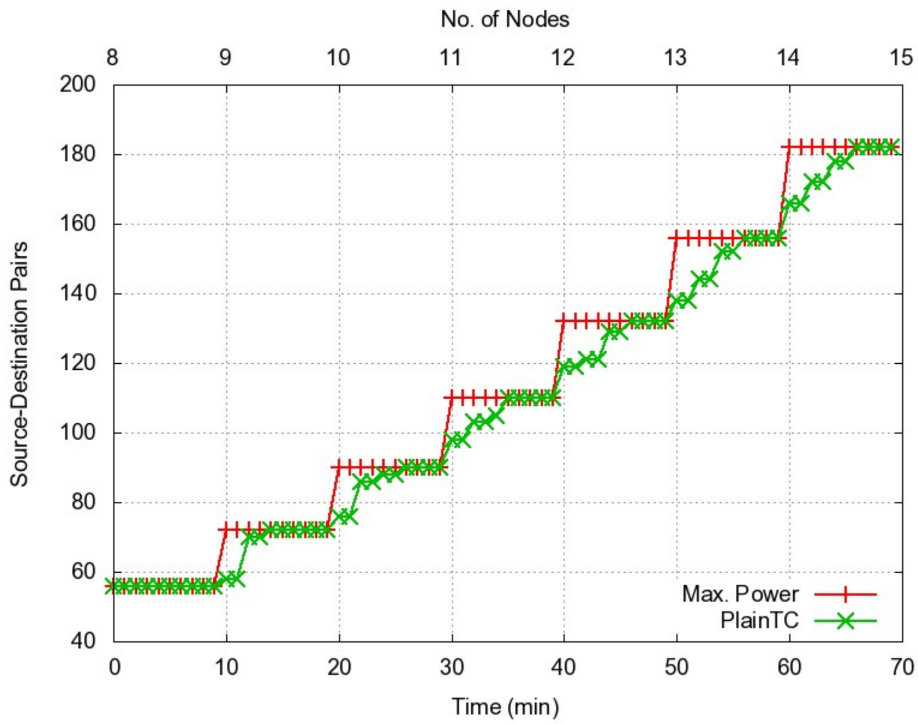



FIGURE 6.1: Dynamic Establishment of Network Connectivity by PlainTC

### 6.2.2 Network Connectivity

The aim of this experiment is to determine the delay in the re-establishment of full network connectivity when a new node is added to the I-WMN. Figure 6.1 shows that PlainTC (in conjunction with the routing protocol) does not instantaneously re-establish full network connectivity when new nodes join the backbone network. A delay is clearly seen to be experienced until full network connectivity can be re-established. The establishment of full network connectivity is shown to be a gradual process. This process occurs whilst the transceiver power levels of the backbone nodes are stabilising and the topology information is being propagated by the OLSR routing protocol.

This delay is a result of the adjustment of node transceiver powers that is required to achieve the desired CNN. Figure 6.2 shows that there is a gradual downwards-adjustment towards the required CNN value. This is a result of the iterative adjustment process followed by PlainTC so as to minimise disruptions to the network.

Figure 6.1 also shows that the delay until full network connectivity is re-established rises as the network size increases. The introduction of the first additional backbone

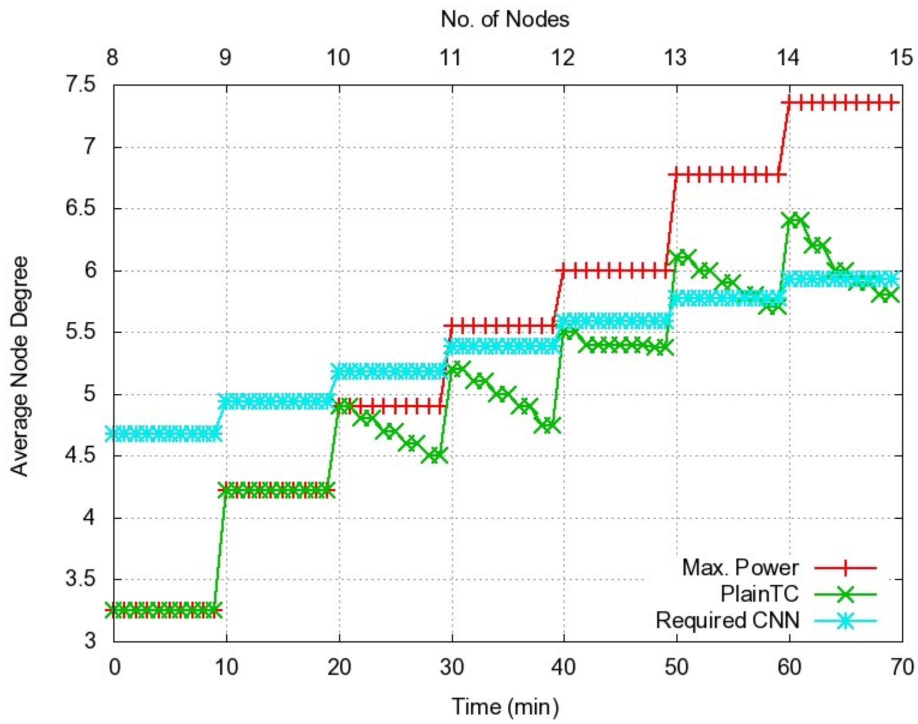



FIGURE 6.2: Dynamic Maintenance of Required CNN by PlainTC

node resulted in an average delay of 2 minutes whilst the addition of the last additional node resulted in an average connectivity delay of 6 minutes. This is a result of the delays incurred by the OLSR routing protocol in the propagation of updated topology information to ever-increasing numbers of network nodes as well as the topology instability caused by PlainTC whilst adjusting node transceiver powers. (This phenomenon is considered in Section 6.2.4).

The rapid increase in the observed delay until network connectivity is re-established is detrimental to the scalability of PlainTC. The unavailability of data in the literature that relates to the time taken to establish network connectivity makes it difficult to assess whether the rapid rise in the delay observed for PlainTC is representative of only CNN-based TC schemes or would also be observed for TC schemes that employ an alternate connectivity strategy.

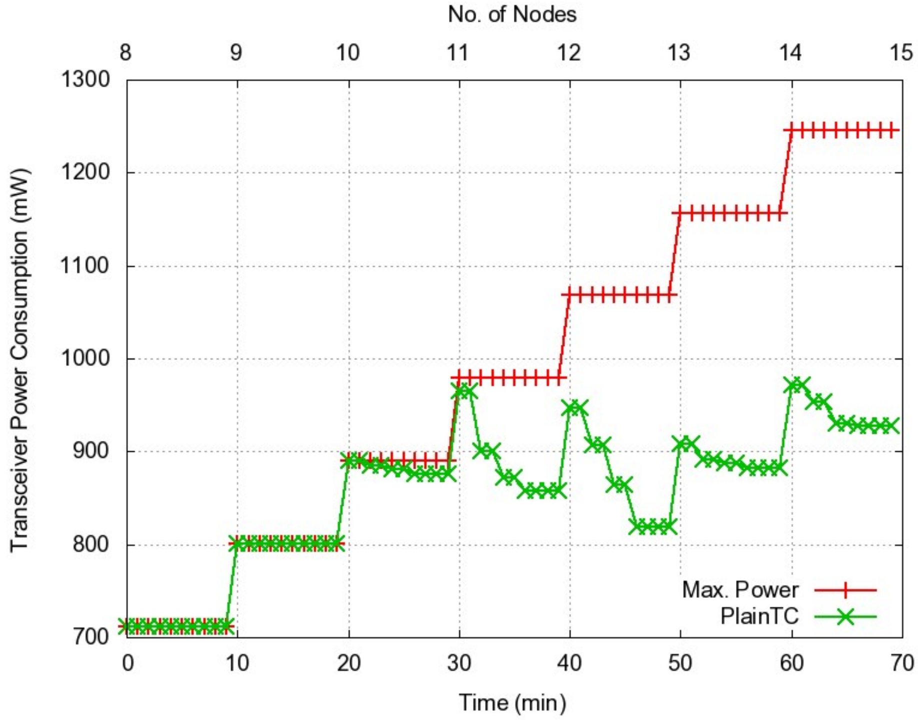



FIGURE 6.3: Dynamic Adjustment of Transceiver Power by PlainTC

### 6.2.3 Transceiver Power Output

This experiment observes the effect that the adjustment of the transceiver power levels of individual nodes has on the cumulative transceiver power output of the I-WMN. The transceiver power adjustment process is driven by the need to maintain the required CNN and is shown in Figure 6.3. The transceiver power adjustment process only begins to produce cumulative transceiver power savings as the network size grows.

The sharp spikes in the cumulative transceiver power output of the backbone network signify the addition of a new node (that is utilising its maximum transceiver power level) to the network and there is also an observed delay until the optimal transceiver power level is reached by this new node. This delay is due to the iterative transceiver power adjustment process that is followed by PlainTC.

The cumulative transceiver power savings reported here are not produced by every network node. Only four (4) nodes were observed to have reduced their transceiver power output levels. This is a result of the number of neighbours required to be maintained by the CNN connectivity strategy employed. These four nodes were found to initially possess neighbourhood sizes that exceeded those prescribed by PlainTC's connectivity strategy and these nodes subsequently gradually reduced their individual transceiver powers until the prescribed neighbourhood size was achieved.

The results shown in Figure 6.3 represent a macro-level view of a relatively stable cumulative transceiver power output that is gradually adjusted downwards by PlainTC. Despite the lack of comparative test-bed-based data, this behaviour is expected of a CNN-based TC scheme. This macro-level viewpoint is informative in showing PlainTC's capacity to produce gradual, cumulative transceiver power savings and the two subsequent experiments in this section will provide a micro-level viewpoint by further analysing the data collected from the test-bed.

### 6.2.4 Effect on the OLSR routing protocol

OLSR is a well-known routing protocol for the I-WMN. Several studies in the literature have analysed OLSR and reported on its performance. However, no previous study, to the best of our knowledge, has employed OLSR in conjunction with a TC scheme.

This experiment observes the side-effects of PlainTC's iterative transceiver power adjustment process. These side-effects are exposed by the metrics measured by the OLSR routing protocol and can be thought of as depicting a micro-level view of PlainTC's iterative transceiver power adjustment process. Figure 6.4 depicts the almost constant oscillation in the number of neighbours maintained by a network node. This is a source of network instability.

Instability in the I-WMN is usually discussed from a network routing perspective where link quality fluctuations and frequent route flapping are observed in deployed networks such as in Ramachandran et al. (2007). However, such behaviour does not account for the almost constant variations in the number of neighbours being observed via OLSR. The changes in the number of neighbours can only be caused by changes in a node's transceiver power levels and these power levels are under the direct control of PlainTC's transceiver power adjustment process. The effect of the almost constant variations in the number of neighbours is topology and neighbourhood instability.

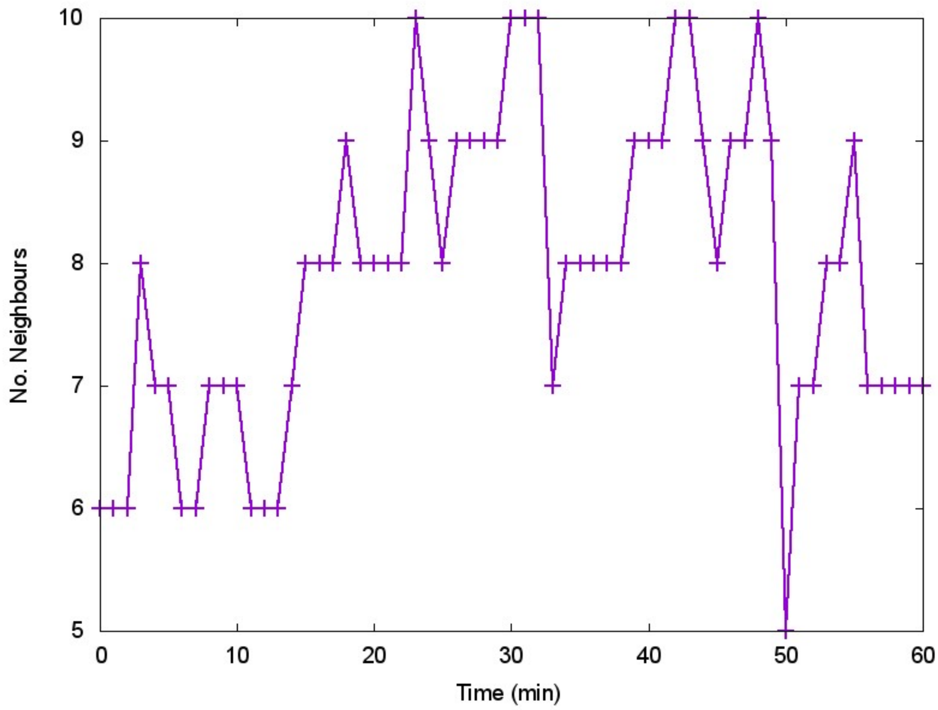



FIGURE 6.4: Varying Neighbourhood Sizes as a result of Oscillating Transceiver Power Levels

The topology and neighbourhood instabilities are a product of the distributed nature of PlainTC. Backbone nodes are independently adjusting their transceiver power outputs which causes their neighbouring nodes to adjust their own transceiver power outputs in response. Thus, a chain-reaction of transceiver power adjustments is observed at the micro level of individual nodes whilst the macro, network-wide view remains stable. The node-level instability negatively affects the operation of OLSR's ETX routing metric as paths to destinations are affected by the changing ETX values experienced by a node as depicted in Figure 6.5. Link quality is known to be subjected to natural variations but these variations are exacerbated by the instabilities caused by PlainTC's transceiver power adjustment process.

### 6.2.5 Network Instability

According to Boushaba et al. (2013), wireless ad-hoc networks are inherently unstable due to the high bandwidth demands and dynamic traffic variations. Therefore, this

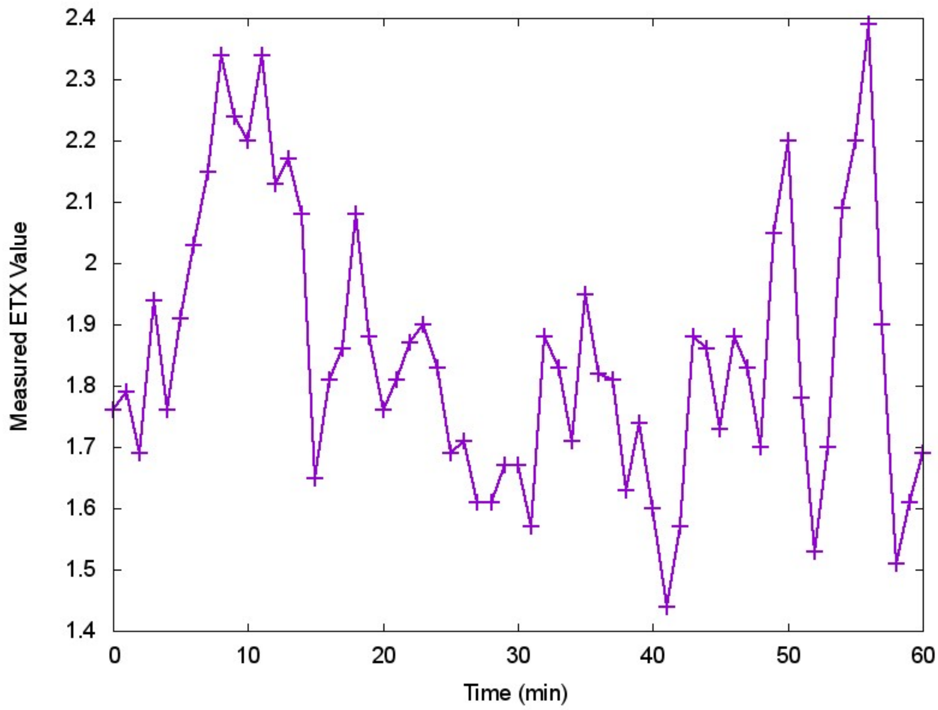



FIGURE 6.5: Varying Link Quality as a result of Oscillating Transceiver Power Levels

particular experiment was carried out without data traffic being transmitted so as to reduce the amount of natural network instability being measured.

The previous experiment showed that the use of PlainTC, which is based upon gradual transceiver power adjustment, injects instability into the network. The instability caused by PlainTC is explored further and differs from the existing literature in that the instability being observed is caused by a TC scheme rather than being reacted to by a routing protocol such as in Ramachandran et al. (2007) and Boushaba et al. (2013).

This experiment was conducted over a 24-hour period to track the changes that occur in a I-WMN that employs PlainTC. The experiment specifically logs the number of changes to important node variables such as the transceiver power output level, neighbourhood size, network size and link quality and represents a micro-level view of the network. These variables are chosen for observation as they represent the context of a node and are easily collected during the normal operation of a test-bed node. Thus, the collection of the values associated with these variables does not place additional demands on resource-constrained I-WMN nodes. This experiment presents a unique look at the long-run behaviour of a TC scheme on a I-WMN test-bed.

The number of changes of each variable is summed across all the network nodes and these changes are presented for twenty-four one-hour intervals. As stated earlier, the 14-node test-bed was employed with no data traffic being sent as the objective is to explore the global effect on network instability of PlainTC's distributed transceiver power adjustment process.

The number of changes recorded for the four node parameters is recorded in Table 6.1. The first hour records the greatest cumulative number of changes for the four variables. This is due to the nodes adjusting their individual transceiver power levels (from the initial maximum power level) to satisfy the required CNN. It would be expected for the subsequent one-hour intervals to show little or no further changes in transceiver power output levels, neighbourhood size and network size as the network topology should have stabilised within the first hour. This is clearly not the case.

Nodes are shown to be constantly adjusting their transceiver power output levels throughout the 24-hour period. This is caused by nodes reacting to the transceiver power changes effected by their neighbours. Changes in transceiver power output levels made by neighbouring nodes cause a change in the neighbourhood size recorded by the affected node. The affected node responds by adjusting its own transceiver power output level. This action, in turn, causes the neighbours of the affected node to adjust their own transceiver power output levels in response. Thus a cycle of transceiver power output level adjustments and counter-adjustments takes place amongst the backbone nodes.

The cascading effect of changes in the transceiver power output causes changes in the other observed variables. The data contained in Table 6.1 shows that changes in transceiver power output levels have a multiplier effect on the other observed variables. Consider a node, node $A$, and its immediate neighbours. $A$ adjusts its transceiver power output level. This change causes a change in its neighbourhood size (in order to maintain the required CNN) and link quality. The neighbours of $A$ may record changes in their own neighbourhood sizes, network sizes and link quality until the local topology stabilises. Note that $A's$ neighbours have not had to adjust their own transceiver power output levels in order to experience changes to their own observed variables. This example shows that a single change in transceiver power output can cause changes in

TABLE 6.1: Number of changes in node variables over a 24-hour period

| Time Period | No. changes in Transceiver Power | No. changes in Neighbourhood Size | No. changes in Network Size | No. changes in Link Quality |
|---|---|---|---|---|
| 08:00-08:59 | 107 | 239 | 303 | 364 |
| 09:00-09:59 | 73 | 102 | 178 | 319 |
| 10:00-10:59 | 70 | 106 | 231 | 304 |
| 11:00-11:59 | 71 | 96 | 220 | 295 |
| 12:00-12:59 | 70 | 104 | 221 | 323 |
| 13:00-13:59 | 69 | 102 | 225 | 302 |
| 14:00-14:59 | 67 | 104 | 220 | 303 |
| 15:00-15:59 | 60 | 87 | 207 | 275 |
| 16:00-16:59 | 70 | 107 | 230 | 319 |
| 17:00-17:59 | 68 | 99 | 221 | 317 |
| 18:00-18:59 | 71 | 97 | 219 | 304 |
| 19:00-19:59 | 69 | 102 | 223 | 314 |
| 20:00-20:59 | 70 | 100 | 227 | 303 |
| 21:00-21:59 | 67 | 97 | 218 | 321 |
| 22:00-22:59 | 69 | 102 | 229 | 309 |
| 23:00-23:59 | 76 | 114 | 235 | 375 |
| 00:00-00:59 | 67 | 93 | 213 | 299 |
| 01:00-01:59 | 67 | 100 | 223 | 312 |
| 02:00-02:59 | 63 | 88 | 205 | 281 |
| 03:00-03:59 | 66 | 91 | 211 | 296 |
| 04:00-04:59 | 68 | 93 | 212 | 296 |
| 05:00-05:59 | 73 | 107 | 220 | 327 |
| 06:00-06:59 | 67 | 94 | 202 | 286 |
| 07:00-07:59 | 65 | 91 | 207 | 289 |
| | | | | |
| **Total** | **1683** | **2515** | **5300** | **7433** |

neighbourhood size, network size and link quality amongst the originating node and each of its immediate neighbours. Thus, changing the transceiver power level causes multiple changes to be recorded for the other observed variables.

The observed changes contribute to instability in the network topology. This topology instability affects the QoS offered by the backbone network. This is because the frequent adjustment of transceiver power output levels causes changes in neighbourhood size. The changes in neighbourhood size affect the routes being maintained by the OLSR routing protocol as these neighbouring nodes are the potential next hop to an intended destination node. Constant changes to the neighbourhood size thus result in changes to the routing table maintained by OLSR. This, in turn, means that routes are constantly changing which causes an increase in OLSR's route creation and maintenance traffic. This additional routing protocol overhead consumes bandwidth at the expense of data traffic and thus the data throughput rates would be reduced in a network that is forwarding traffic.

It must be remembered that the instability measured in this experiment only captured the instabilities caused by the TC scheme as the network was not carrying data traffic. Other works in the literature have shown the instability caused by traffic forwarding in the I-WMN. Thus, PlainTC's use in a deployed I-WMN is likely to exacerbate the existing network instability. Solutions to the instability caused by traffic forwarding have been proposed in Ramachandran et al. (2007) and Boushaba et al. (2013) in the form of stability-aware routing protocols and a similar solution must be proposed for the instability caused by PlainTC and other CNN-based TC schemes. Chapter 7 proposes a context-based solution tailored to the peculiarities of the deployed network.

The next set of experiments resembles those most commonly seen in the literature and is used to assess PlainTC's performance using more traditional, well-known performance metrics and measurement methodologies.

## 6.3 A Traditional Evaluation of PlainTC

This section, using the traditional static snapshot method, evaluates PlainTC's ability to achieve and/or maintain the desirable network properties commonly associated with TC in the literature. It is widely accepted in the literature that TC is an effective mechanism for providing multi-faceted benefits to the wireless ad-hoc networks that employ such schemes. Common properties associated with TC in the literature include, but are not limited to the capacity to:

- maintain network connectivity
- reduce cumulative transceiver power consumption

- improve energy efficiency
- reduce interference
- enhance spatial re-use
- improve network capacity
- lengthen network lifetime (if nodes are battery-powered)

Apart from the properties, there are some implicit assumptions about TC schemes and the nature of the topologies they create when eventually used in wireless ad-hoc and mesh networks. These assumptions include, but are not limited to the assumption that:

- TC will, by default, create a stable network topology
- TC will result in transceiver power savings at each network node

The evaluation contained in this section assesses PlainTC against these commonly associated properties and assumptions of TC schemes. The evaluation has been conducted on both an indoor I-WMN test-bed and a two-phase simulator that have been described in Chapter 5. The data collection methodology employed for the performance metrics being evaluated can also be found in Chapter 5.

The PlainTC scheme presented in Chapter 4 is compared with the Max. Power scheme. The Max. Power scheme presents the current status quo in deployed WMNs and is therefore utilised as a benchmark for PlainTC. Each subsequent experiment explores the ability of the CNN-based PlainTC scheme to achieve an afore-mentioned property associated with TC schemes.

### 6.3.1 Network Connectivity

The maintenance of network connectivity is a primary objective of TC in ad-hoc and mesh networks. Thus, in this subsection, the ability of PlainTC to maintain network connectivity is evaluated. Network connectivity is best measured at the Network Layer,

TABLE 6.2: Network Connectivity (Indoor Test-bed)

| Network Size | Src-Dest Pairs (Max. Power) | Src-Dest Pairs (PlainTC) |
|---|---|---|
| 8 | 56 | 56 |
| 9 | 72 | 72 |
| 10 | 90 | 90 |
| 11 | 110 | 110 |
| 12 | 132 | 132 |
| 13 | 156 | 155 |
| 14 | 182 | 180 |

where a global view of the network is possible. Thus, the number of available source-destination pairs is taken as a measure of the network connectivity. The greater the number of source-destination pairs, the more connected the network and vice-versa.

The Max. Power scheme assures network connectivity, resulting in $n^2 - n$ source-destination pairs where $n$ refers to the number of nodes in the I-WMN. The PlainTC scheme uses the probabilistic CNN method to maintain network connectivity. I-WMN nodes are required to maintain a given number of one-hop neighbours. PlainTC's use of the CNN by the individual nodes is shown in both the test-bed and simulation-based evaluations (see Table 6.2 and Table 6.3) to be a competent network connectivity method.

The route redundancy promoted by PlainTC's use of the CNN connectivity method ensures that routes exist to nearly all possible destinations.

The proficiency of the network connectivity strategy employed by PlainTC was also shown in Xue and Kumar (2004) to produce asymptotic connectivity approaching 1 as the network size grows. This achievement of close to 100% connectivity is also consistent with the connectivity result obtained in Srivastava et al. (2004) for the node-degree-based LINT protocol.

### 6.3.2 Transceiver Power Savings

The secondary objective of TC in ad-hoc and wireless mesh networks is to produce cumulative transceiver power savings. These cumulative transceiver power savings,

TABLE 6.3: Network Connectivity (Simulation)

| Network Size | Src-Dest Pairs (Max. Power) | Src-Dest Pairs (PlainTC) |
|---|---|---|
| 20 | 380 | 380 |
| 40 | 1560 | 1560 |
| 60 | 3540 | 3539 |
| 80 | 6320 | 6317 |
| 100 | 9900 | 9898 |
| 120 | 14280 | 14277 |

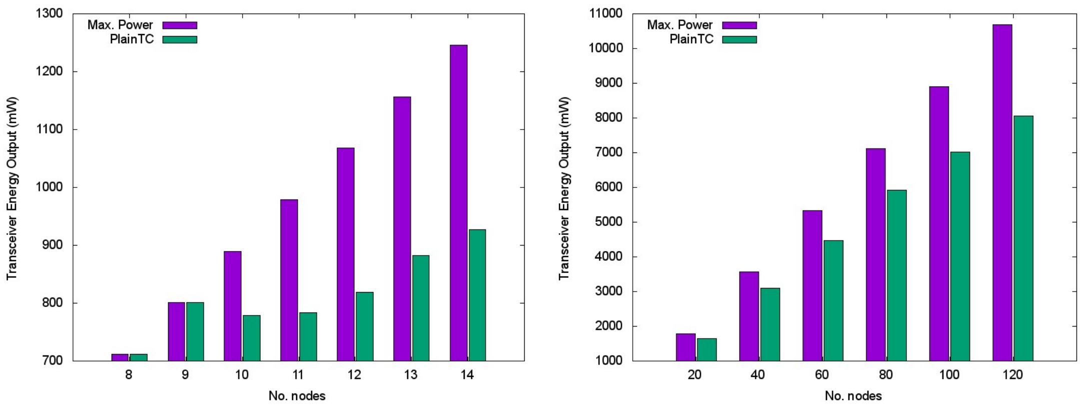
TABLE 6.4: Percentage Power Savings Achieved (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 0 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 12.5 |
| 11 | 0 | 20 |
| 12 | 0 | 23.3 |
| 13 | 0 | 22 |
| 14 | 0 | 25.6 |

if achieved, are believed to improve the energy efficiency of the network since data transmission consumes less energy.

Both the test-bed (see Figure 6.6(a) and Table 6.4) and simulation-based (see Figure 6.6(b) and Table 6.5) evaluations indicate that PlainTC is capable of producing cumulative transceiver power savings. PlainTC benefits from not having to maintain the original one-hop neighbourhood of a node as the CNN to be maintained is less than the original neighbourhood size. Both Table 6.4 and Table 6.5 indicate that greater cumulative transceiver power savings are achieved as the network size increases. This occurs because the CNN method causes the number of neighbours to be maintained to rise at a slower rate than increases in the network size.

Transceiver power savings were also reported for the $\text{FLSS}_\text{k}$ and TAP schemes in Li and Hou (2006) and Hu et al. (2006), respectively. $\text{FLSS}_\text{k}$ with $\text{k} = 3$ is a node-degree-based TC scheme that was reported to produce power savings of 13% in a 120 node network. TAP was found to produce power savings of 24% in a 120 node network by considering traffic patterns and local connectivity.




(a) Indoor Test-bed

(b) Simulation

FIGURE 6.6: Transceiver Power Consumption

TABLE 6.5: Percentage Power Savings Achieved (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 0 | 7.8 |
| 40 | 0 | 13.1 |
| 60 | 0 | 16.1 |
| 80 | 0 | 16.7 |
| 100 | 0 | 21.2 |
| 120 | 0 | 24.6 |

### 6.3.3 Source of Transceiver Power Savings

Previous studies have generally assumed that transceiver power savings will be realised. The results presented in the previous experiment together with the works reported in Li and Hou (2006) and Hu et al. (2006) have actually quantified these power savings but these previous studies did not determine whether each individual network node was contributing to these power savings.

In this experiment, the transceiver power output levels produced by the individual nodes as a result of PlainTC are investigated further. This represents, to the best of our knowledge, a unique look at the post-TC transceiver powers associated with all of the nodes in the backbone network. A listing of the individual transceiver power outputs for the testbed is contained in Table 6.6. In addition, the distances of the various nodes from an imaginary central point can be found in the table.

Table 6.6 indicates that only four (4) nodes are responsible for the cumulative transceiver power savings shown in Figure 6.6(a). These four nodes produced significant cuts to their transceiver power levels and are centrally located which seems to suggest that node position within the I-WMN influences the resultant transceiver power output level of the node.

The testbed comprises arbitrarily-placed nodes so the placement of the nodes may be unduly affecting their ability to reduce their transceiver power levels. The potential influence of centrally-located nodes was first explored in Souihli et al. (2009) in the context of routing table sizes. The simulation results depicted in Figure 6.7 (from uniformly distributed backbone nodes) show that centrally-located nodes achieve greater transceiver power output savings than nodes closer to the network edge. Similar behaviour for other CNN connectivity schemes and node distributions are found in Section 3.2[1].

Centrally-located nodes possess more neighbours than their outlying peers[2]. Thus, the required number of neighbours to be maintained can be achieved by using lower transceiver power levels. Smaller network sizes result in outlying nodes possessing too few neighbours to meet the CNN threshold. This results in the outlying nodes using the maximum transceiver power output level in a futile attempt to maintain the specified CNN. The percentage of outlying nodes that employ the maximum transceiver power output level is found to decrease as the network size increases (see Table 6.7) due to an increase in the number of neighbours in the outlying regions of the network.

Table 6.7 also shows that PlainTC's CNN connectivity strategy prevents drastic reductions in transceiver power output level. Hence, there are no backbone nodes employing transceiver power output levels in the 1% to 20% range. In addition, the percentage of nodes employing transceiver power output levels in the 21% to 70% range are minimal. This is a result of the PlainTC CNN connectivity strategy's emphasis on achieving network connectivity by maintaining $\lceil 5.1774log(n) \rceil$ neighbours.

[1]Section 3.2 evaluated the ability of connectivity strategies to produce cumulative transceiver power savings

[2]This finding was re-iterated in Souihli et al. (2009) by showing that centrally-located nodes possessed a larger MPR selector set in networks using the OLSR protocol.

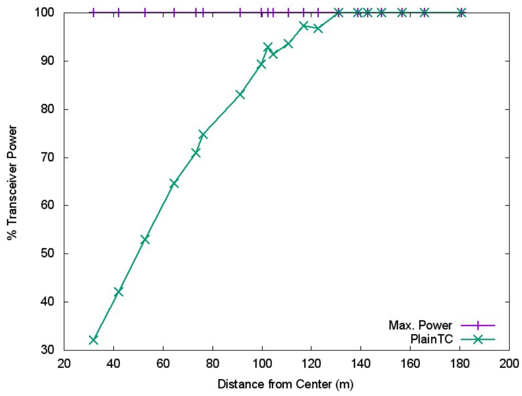
TABLE 6.6: Transceiver Power Output vs. Distance from Network Center (Indoor Test-bed)

| Node ID | Distance from Center (m) | Max. Power (mW) | PlainTC (mW) |
|---|---|---|---|
| 2 | 2.42 | 89 | 89 |
| 3 | 2.58 | 89 | 89 |
| 4 | 2.70 | 89 | 89 |
| 5 | 2.34 | 89 | 89 |
| 6 | 1.31 | 89 | 89 |
| 7 | 1.35 | 89 | 1 |
| 8 | 0.64 | 89 | 1 |
| 9 | 2.56 | 89 | 89 |
| 10 | 2.69 | 89 | 89 |
| 11 | 2.71 | 89 | 89 |
| 12 | 1.27 | 89 | 1 |
| 13 | 0.39 | 89 | 89 |
| 14 | 2.56 | 89 | 89 |
| 15 | 1.38 | 89 | 1 |

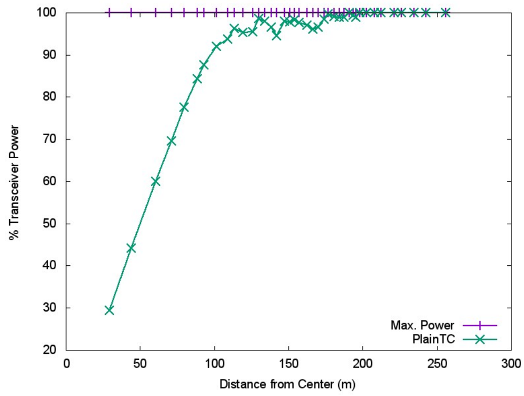
The majority of backbone network nodes are found to employ transceiver power output levels between 71% and 99% of the maximum power output level. Thus, the majority of nodes produce between 1% and 29% in transceiver power output savings. Having the majority of nodes using between 71% and 99% of the maximum transceiver power output level improves the possibility of achieving or maintaining network connectivity since the only known guarantor of connectivity in deployed WMNs is the use of the maximum transceiver power level.

### 6.3.4 Path Length

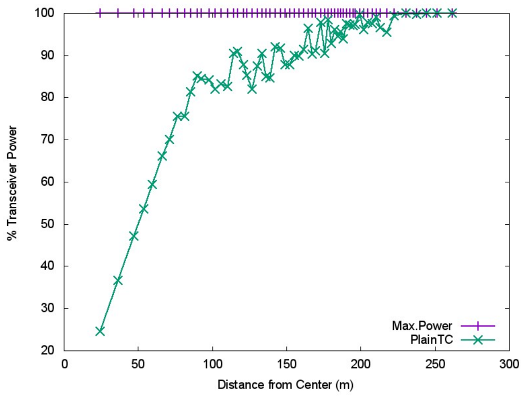
A reduction in the transceiver power output level of a node causes a reduction in its communication range. This is thought to result in longer average path lengths since a greater number of intermediate nodes are required to forward network traffic to the intended destinations. In this subsection, the average path lengths produced by PlainTC are explored. Two common communication scenarios experienced in deployed WMNs are considered.



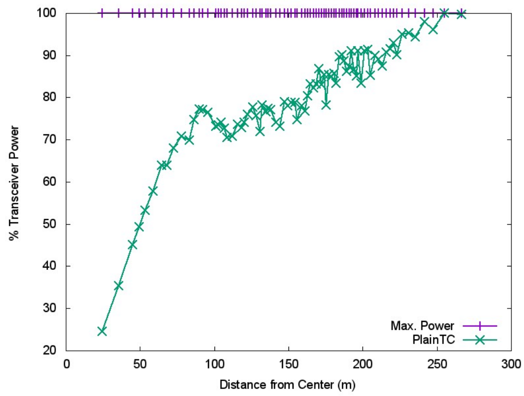

(a) 20 node network



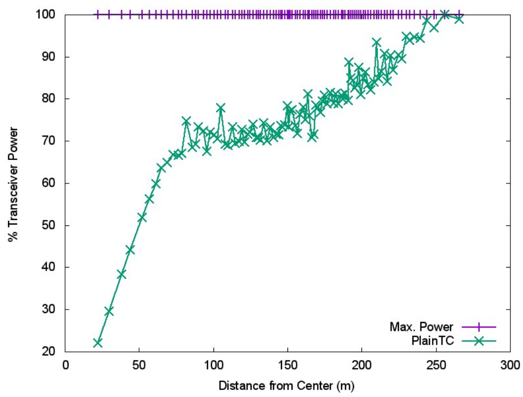

(b) 40 node network



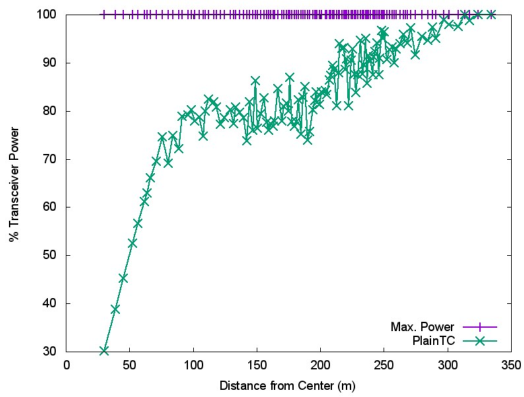

(c) 60 node network




(d) 80 node network




(e) 100 node network




(f) 120 node network

FIGURE 6.7: Transceiver Power Output vs. Distance from Network Center (Simulation)

TABLE 6.7: Percentage distribution of backbone nodes amongst the various transceiver power output bands

| Transceiver Power Level Bands (%) | Network Size | | | | | |
|---|---|---|---|---|---|---|
| | 20 | 40 | 60 | 80 | 100 | 120 |
| 1-10 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11-20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21-30 | 0 | 2.5 | 1.67 | 1.25 | 2 | 0.83 |
| 31-40 | 5 | 0 | 1.67 | 1.25 | 1 | 0.83 |
| 41-50 | 5 | 2.5 | 1.67 | 2.5 | 1 | 0.83 |
| 51-60 | 5 | 2.5 | 3.34 | 2.5 | 3 | 1.67 |
| 61-70 | 5 | 2.5 | 3.34 | 8.75 | 19 | 4.17 |
| 71-80 | 10 | 2.5 | 5.01 | 37.5 | 40 | 24.17 |
| 81-90 | 10 | 5 | 26.72 | 27.5 | 23 | 35.83 |
| 91-99 | 25 | 62.5 | 50.1 | 16.25 | 9 | 29.14 |
| 100 | 35 | 22.5 | 7 | 2.5 | 1 | 3.3 |

Deployed WMNs support both Intra-mesh and Portal-oriented communications. Intra-mesh communications are forwarded along the backbone and never exit the mesh network because both the source and destination exist within the network. Portal-oriented communications involve the Mesh Portal Point. Traffic is either intended for a destination outside of the mesh network or the traffic originates from an external source but is intended for an internal destination. Both communication types are depicted in Figure 5.2.

PlainTC is found to increase the average path length on both the test-bed and the simulator for both communication scenarios (see Tables 6.8, 6.9, 6.10 and 6.11). The test-bed results for the two communication scenarios show a minor increase in the average path length. This is a result of the majority of test-bed nodes continuing to operate with the maximum transceiver power output level. Section 6.3.3 showed that only four nodes were found to reduce their transceiver power output levels and this did not cause a significant change in the forwarding paths employed.

The simulation results for both communications scenarios found in Tables 6.9 and 6.11 show a significant increase in the average path lengths produced by PlainTC. The transceiver power output reductions achieved by the majority of network nodes (refer to Section 6.3.3) results in a re-arrangement of the network topology caused by the reduced communication ranges of the network nodes. The re-arranged network topology

TABLE 6.8: Average Path Lengths Achieved for Intra-mesh Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 2.1 | 2.1 |
| 9 | 2.2 | 2.2 |
| 10 | 2.3 | 2.4 |
| 11 | 2.5 | 2.7 |
| 12 | 2.5 | 3.0 |
| 13 | 2.7 | 2.9 |
| 14 | 2.7 | 2.9 |

TABLE 6.9: Average Path Lengths Achieved for Intra-mesh Traffic (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 2.9 | 4.8 |
| 40 | 4.7 | 5.8 |
| 60 | 5.8 | 6.9 |
| 80 | 6.5 | 7.4 |
| 100 | 6.8 | 8.8 |
| 120 | 6.9 | 10.4 |

TABLE 6.10: Average Path Lengths Achieved for Portal-oriented Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 1.9 | 1.9 |
| 9 | 2.1 | 2.1 |
| 10 | 2.1 | 2.3 |
| 11 | 2.4 | 2.5 |
| 12 | 2.4 | 2.6 |
| 13 | 2.5 | 2.7 |
| 14 | 2.5 | 2.7 |

requires new forwarding paths and these forwarding paths are longer because of the reduced communication ranges of the network nodes. The forwarding paths are found to be between 1-3 hops longer after employing PlainTC.

This study has considered the path length for two differing communication scenarios. Other works in the literature only consider the intra-mesh traffic scenario and do not explicitly measure the increase in path lengths that their proposed TC schemes cause. The literature tends to graphically depict increases in path lengths by showing the pre-TC network topology and the sparser, post-TC topology. The sparser network topology

TABLE 6.11: Average Path Lengths Achieved for Portal-oriented Traffic (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 2.5 | 4.6 |
| 40 | 4.4 | 5.4 |
| 60 | 5.4 | 6.5 |
| 80 | 6.1 | 7.1 |
| 100 | 6.3 | 8.3 |
| 120 | 6.5 | 9.9 |

implies longer path lengths. To the best of our knowledge, the only other explicit measurement of the average path length for a TC scheme can be found in Li et al. (2005b). The LMST TC scheme proposed in this work reduced the average link length by approximately 100m in a 120-node network. This reduction in link length implies that the average path length, measured by the number of hops or links, was increased by the application of LMST. A similar result was obtained for PlainTC in both the traffic scenarios under consideration.

### 6.3.5 Interference

Interference can be measured at both the MAC Layer as well as the Physical Layer. MAC Layer interference refers to the number of one-hop neighbours that are affected by a node's communications whilst the Physical Layer interference deals with the noise floor (or background noise level) generated by the transceiver powers of the neighbouring nodes. The noise floor is a source of interference as it dictates that the power level used for data transmission must, at the very least, exceed the noise floor (by some specified constant) for the transmission to be successfully decoded by the recipient node.

In this sub-section, the impacts of PlainTC on both the Physical Layer and MAC-level interference are presented.

#### 6.3.5.1 Physical Layer Interference

The impacts of the Max. Power and PlainTC schemes on the Physical Layer interference experienced could not be determined by the simulation-based evaluation due to the

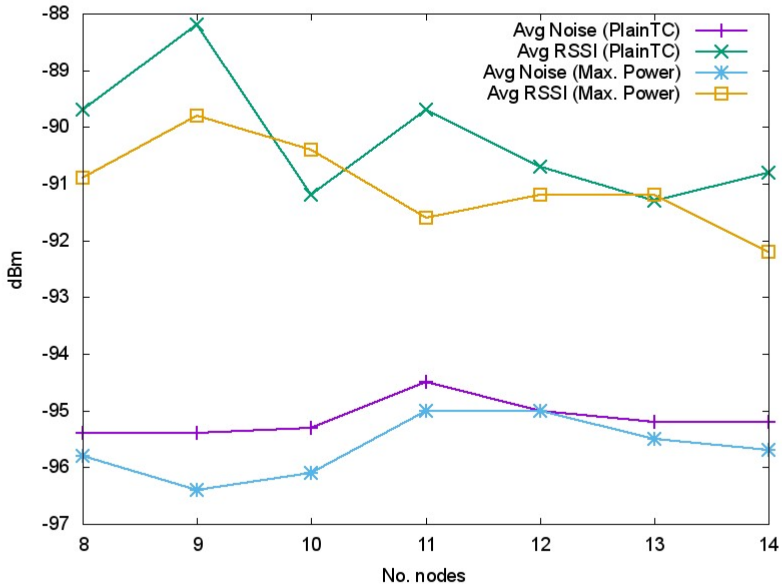



FIGURE 6.8: Physical Layer Interference (Indoor Test-bed)

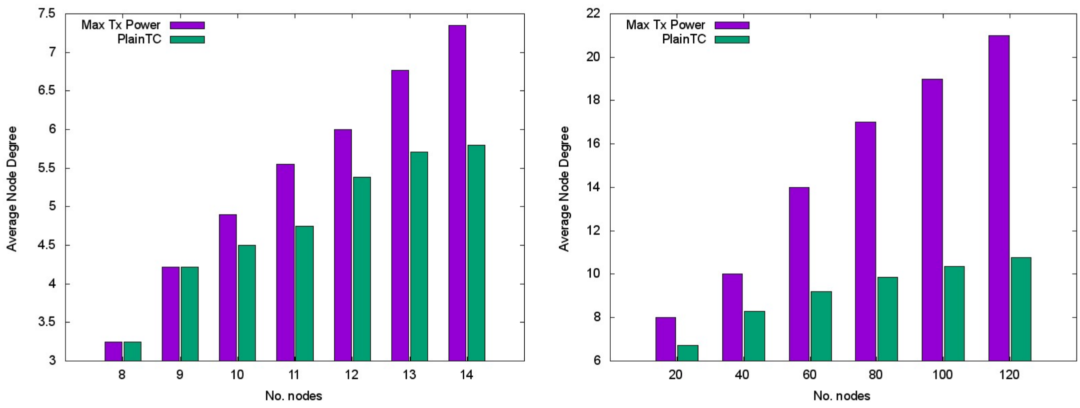
lack of support provided by the ns-2 simulator for RSSI values. Thus, only the results obtained from the test-bed evaluation are presented here.

Despite the cumulative transceiver power output savings produced by PlainTC, there is no observed improvement in the noise and received signal strength levels depicted in Figure 6.8. The noise level is also influenced by the presence of other WiFi networks in the vicinity of the test-bed. This accounts for the lack of improvement in the noise level. The received signal strength shows a marginal improvement caused by the nodes that reduced their transceiver power output levels. However, this was not sufficient to improve the physical layer environment.

To the best of our knowledge, no prior instances of such Physical Layer measurements have been found for a TC scheme.

#### 6.3.5.2 MAC-level Interference

The MAC-level interference of the schemes being evaluated is determined by the average number of neighbours maintained by each network node. This number is also referred to as the *node degree*. The node degree is a reliable indicator of MAC-level interference because it indicates the number of neighbouring nodes that must remain




(a) Indoor Test-bed

(b) Simulation

FIGURE 6.9: Average MAC-level interference

silent when a node communicates. The node degree thus also indicates the number of potentially interfering nodes that need to be contended with.

PlainTC's use of the CNN connectivity strategy produces significant reductions in potential MAC-level interference. This phenomenon can be observed in both the test-bed and simulator results depicted in Figure 6.9. The adaptive CNN and its associated transceiver power output reductions result in increasing reductions in MAC-level interference as the network size increases. Thus, PlainTC is able to significantly reduce contention for the shared communication channel as there are fewer competing backbone nodes. In addition, a node that is granted access to the shared communication channel will be forcing a smaller number of neighbouring backbone nodes to remain silent.

The reductions in MAC-level interference observed in the test-bed are produced by those nodes that reduced their transceiver power output levels[3]. These nodes were shown to be centrally located and initially possessed large one-hop neighbourhoods. PlainTC caused these nodes to reduce the size of their one-hop neighbourhoods in order to maintain the desired CNN value. The simulator produced greater reductions in average MAC-level interference as a greater percentage of network nodes were found in Section 6.3.3 to reduce their transceiver power output levels.

[3]Only four (4) nodes were found to reduce their transceiver power output levels in Section 6.3.3

PlainTC produced a reduction in average node degree of 50% with the simulator. Other simulation-based works have been found to report reductions in node degree for their post-TC networks. The TC schemes found in Aron et al. (2008a); Gao et al. (2008); Aron et al. (2008b); Li and Hou (2006) were found to achieve reductions of 66%, 75%, 75% and 75%, respectively. These schemes employed more aggressive connectivity strategies than that used by PlainTC in order to lower the average number of one-hop neighbours.

### 6.3.6 Spatial Re-use

The reduction in transceiver power output levels produced by TC is often thought to improve the spatial re-use of the wireless communication medium via the reduction in the interference ranges of the backbone nodes. The preceding subsection showed that a reduced interference range produces reduced contention for the shared communication medium. This subsection investigates whether PlainTC possesses the ability to increase the number of concurrent traffic flows that can be successfully initiated and sustained in the backbone network.

A typical I-WMN will support the two communication scenarios previously described in Section 6.3.4 and depicted in Figure 5.2. Both scenarios are considered in this experiment.

#### 6.3.6.1 Intra-mesh Traffic

The spatial re-use achieved is determined by the maximum number of concurrent (simultaneous) traffic flows that can be supported by the backbone network. PlainTC is found to match the performance of the Max. Power scheme in the test-bed (see Table 6.12). Both schemes support the maximum number of concurrent traffic flows for all network sizes under the experiment setup described in Section 5.4.6. This is a result of the similar physical layer environments experienced by both schemes, as shown earlier in Section 6.3.5.1.

TABLE 6.12: Max. number of Concurrent Traffic Flows for the Intra-Mesh Scenario (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 4 | 4 |
| 9 | 4 | 4 |
| 10 | 5 | 5 |
| 11 | 5 | 5 |
| 12 | 6 | 6 |
| 13 | 6 | 6 |
| 14 | 7 | 7 |

TABLE 6.13: Max. number of Concurrent Traffic Flows for the Intra-Mesh Scenario (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 8 | 10 |
| 40 | 15 | 20 |
| 60 | 25 | 28 |
| 80 | 31 | 38 |
| 100 | 41 | 47 |
| 120 | 49 | 54 |

PlainTC does outperform the Max. Power scheme by supporting a greater number of concurrent traffic flows on the network simulator (see Table 6.13). PlainTC benefits from the reduced communication and interference ranges that the majority of backbone nodes are able to achieve[4]. This allows for increased concurrent traffic flows within the backbone network. PlainTC's performance was marginally affected by the absence of routes to some of the intended destinations due to the inability to achieve full network connectivity (see Section 6.3.1).

#### 6.3.6.2 Portal-Oriented Traffic

The Portal-oriented traffic scenario differs from the intra-mesh scenario because the results presented may be seen as a much more accurate reflection of the Mesh Portal's

[4]Refer to Section 6.3.3 for the reductions in the communication ranges and Section 6.3.5.2 for the resultant reduction in the interference range as inferred by the reduction in the MAC-level interference or one-hop neighbourhood size

TABLE 6.14: Max. number of Concurrent Traffic Flows for the Web-based Scenario (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 4 | 4 |
| 9 | 4 | 4 |
| 10 | 5 | 5 |
| 11 | 5 | 5 |
| 12 | 6 | 6 |
| 13 | 6 | 6 |
| 14 | 7 | 7 |

TABLE 6.15: Max. number of Concurrent Traffic Flows for the Web-based Scenario (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 9 | 9 |
| 40 | 18 | 17 |
| 60 | 26 | 25 |
| 80 | 32 | 30 |
| 100 | 41 | 42 |
| 120 | 47 | 51 |

(MPP) capabilities[5] rather than the spatial re-use capabilities of the evaluated schemes. However, the evaluated TC schemes are able to influence the number of concurrent traffic flows that the MPP will be accommodating at any point in time.

The results for this aspect of the experiment can be found in Tables 6.14 and 6.15. PlainTC is able to match the performance of the Max. Power scheme in the test-bed (see Table 6.14). Both schemes support the maximum number of concurrent traffic flows for all network sizes.

The network simulator allows for greater network sizes to be evaluated. PlainTC is found to support a greater number of concurrent traffic flows than the Max. Power scheme in the simulator (see Table 6.15). This is a product of the transceiver power savings produced by the network nodes. The power savings have resulted in reduced interference ranges which allows for a greater number of concurrent traffic flows.

[5]I-WMNs contain at least one Mesh Portal Point through which all traffic enters and exits the network. Thus, all concurrent traffic flows merge at these nodes and the maximum number of traffic flows supported by the MPPs can be taken to be a measure of the spatial re-use capabilities of the network. The MPP is usually a much more capable node with the ability to support more traffic flows than a MAP or a MP.

To the best of our knowledge, no prior instances of measurements of the number of concurrent traffic flows have been found for a TC scheme.

### 6.3.7 Data Traffic Performance

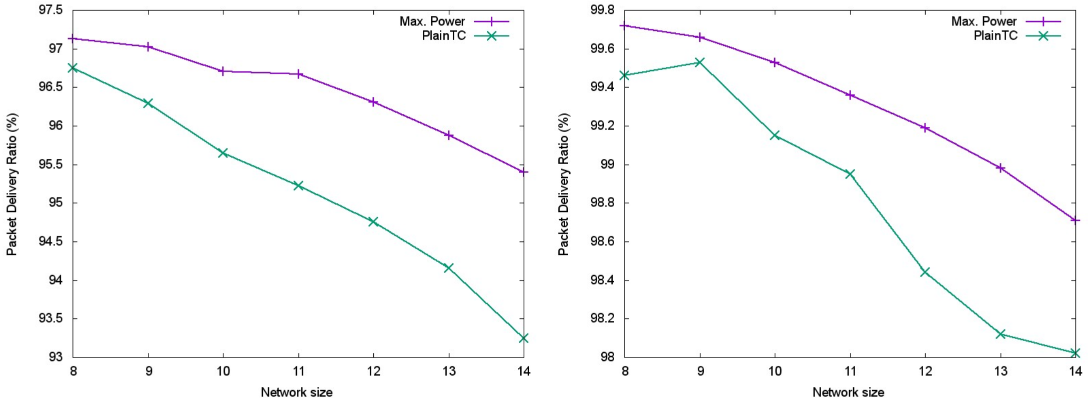
Data traffic performance is an important aspect of the QoS experienced by the traffic flows within a network. The I-WMN supports the two communication scenarios described in Section 6.3.4 and depicted in Figure 5.2. Existing studies of TC's effect on Application Layer traffic performance consider only the Intra-mesh traffic scenario. This subsection considers both the Intra-mesh and Portal-oriented scenarios. The QoS experienced by network traffic flows is commonly measured in the literature using the Packet Delivery Ratio (PDR) and Throughput metrics. Results from both the test-bed and simulator are presented for both communication scenarios.

#### 6.3.7.1 Packet Delivery Ratio (PDR)

The PDR reflects the ability of the network to deliver traffic to its intended destination. The Max. Power scheme outperforms PlainTC for all network sizes on both the test-bed and the network simulator. The test-bed outperformance of Max. Power over PlainTC is however smaller than in the simulator. The test-bed evaluation earlier showed that despite the achievement of cumulative transceiver power savings, the only other observed change achieved by PlainTC was in the average path length. The greater average path length achieved by PlainTC[6] is the cause of the lower achieved PDR values contained in Tables 6.16 and 6.17 and depicted in Figure 6.10. The PDR achieved for the intra-mesh traffic is also affected by the UDP transport protocol employed for such a scenario. Packet losses are higher when using UDP due to its lack of a successful packet receipt acknowledgement mechanism.

The simulator results contained in Tables 6.18 and 6.19 and depicted in Figure 6.11 produce a greater underperformance for PlainTC due to the larger network sizes considered. The substantial increases in the average path length of traffic flows recorded in

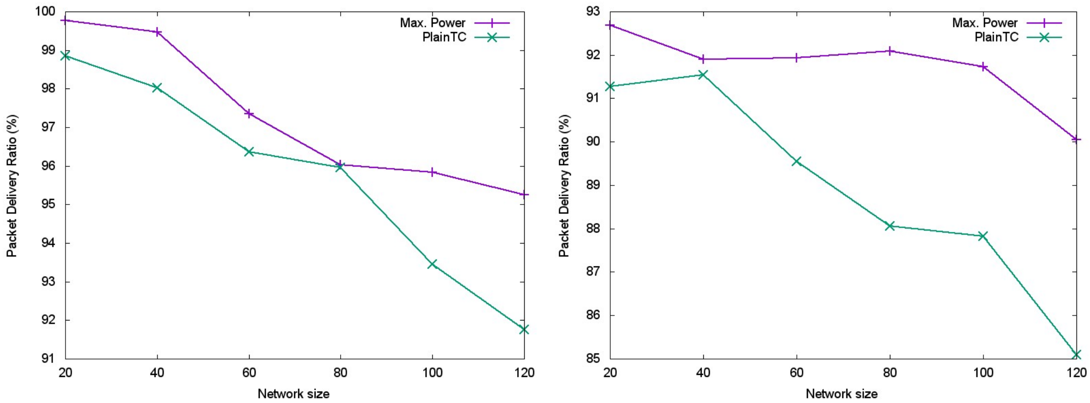
[6]Refer to Section 6.3.4




(a) Intra-mesh traffic

(b) Portal-oriented Traffic

FIGURE 6.10: Packet Delivery Ratio (Test-bed)




(a) Intra-mesh traffic

(b) Portal-oriented Traffic

FIGURE 6.11: Packet Delivery Ratio (Simulation)

Section 6.3.4 contribute to PlainTC's underperformance. In contrast to the test-bed evaluation, the Portal-oriented traffic scenario produces lower PDR values than the Intra-mesh scenario. This is a result of the congestion caused by having all network traffic either being destined for or originating from the Portal node. The use of the TCP transport protocol in such a scenario is known to exacerbate congestion in wireless networks. This results in the continuous dropping of packets with a resultant low PDR, as depicted in Figure 6.11(b).

The high PDR values achieved with the test-bed and the simulator compare favourably

TABLE 6.16: Average PDR Achieved for Intra-mesh Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 97.13 | 96.75 |
| 9 | 97.02 | 96.29 |
| 10 | 96.71 | 95.65 |
| 11 | 96.67 | 95.22 |
| 12 | 96.31 | 94.76 |
| 13 | 95.88 | 94.16 |
| 14 | 95.4 | 93.25 |

TABLE 6.17: Average PDR Achieved for Portal-oriented Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 99.72 | 99.46 |
| 9 | 99.66 | 99.53 |
| 10 | 99.53 | 99.15 |
| 11 | 99.36 | 98.95 |
| 12 | 99.19 | 98.44 |
| 13 | 98.98 | 98.12 |
| 14 | 98.71 | 98.02 |

TABLE 6.18: Average PDR Achieved for Intra-mesh Traffic (Simulator)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 99.77 | 98.85 |
| 40 | 99.47 | 98.02 |
| 60 | 97.3 | 96.35 |
| 80 | 96.02 | 95.95 |
| 100 | 95.82 | 93.45 |
| 120 | 95.25 | 91.75 |

TABLE 6.19: Average PDR Achieved for Portal-oriented Traffic (Simulator)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 92.69 | 91.27 |
| 40 | 91.9 | 91.54 |
| 60 | 91.92 | 89.54 |
| 80 | 92.09 | 88.05 |
| 100 | 91.72 | 87.81 |
| 120 | 90.05 | 85.08 |

with existing results reported in the literature. The work in Valera et al. (2008) is a test-bed-based study and reports wide divergences in the PDR values measured at various transceiver power output levels. However the majority of links produce PDR values close to 100%. Simulation-based PDR results can be found in Kawadia and Kumar (2005) and (Kadivar et al., 2009). The CLUSTERPOW scheme reported in Kawadia and Kumar (2005) was found to improve on the PDR achieved at max. power and a PDR value close to 100% was obtained. The OTTC and XTC schemes reported on in Kadivar et al. (2009) were found to achieve PDR values of 93% and 90%, respectively.

#### 6.3.7.2 Throughput

Throughput determines the rate at which data is received at the destination and can be seen as an indicator of the QoS levels on offer by the networks subjected to the evaluated TC schemes. Both the intra-mesh and portal-oriented traffic scenarios are considered.

PlainTC is found to underperform the Max. Power scheme in both the test-bed and simulator for the intra-mesh scenario with both schemes producing reduced average throughputs as the network size increases. The resultant performance of both schemes is contained in Tables 6.20 and 6.21. The test-bed underperformance of PlainTC is however relatively small (see Figure 6.12(a)) and is caused by the slightly longer average path length (see Section 6.3.4).

The larger network sizes considered in the simulator shows a greater underperformance of PlainTC when compared to the Max. Power scheme. Figure 6.13(a) depicts a narrowing in the difference between the average throughputs of the considered schemes. The Max. Power scheme benefits from the shorter path lengths at the lower network sizes but suffers from increases in the MAC-level interference at the larger network sizes as discussed in Section 6.3.5.2. PlainTC however, suffers from longer average path lengths and PDRs for all simulated network sizes (see Sections 6.3.4 and 6.3.7.1 respectively) but benefits from the greater reduction in MAC-level interference and spatial reuse as the network size increases (see Sections 6.3.5.2 and 6.3.6 respectively).

The Portal-oriented traffic scenario is also considered. Both the Max. Power and PlainTC schemes achieve lower average throughputs in this scenario when compared to the intra-mesh scenario. This is a result of the network traffic converging upon the Mesh Portal (MPP) and the associated traffic bottleneck that arises. The performance of both schemes is contained in Tables 6.22 and 6.23 and depicted in Figures 6.12(b) and 6.13(b).

PlainTC is found to underperform the Max. Power scheme in both the test-bed and simulator with both schemes producing lower average throughputs as the network size increases. PlainTC's underperformance on the test-bed is relatively minor (see Figure 6.12(b)) and is influenced by the longer average path lengths reported in Section 6.3.4. The differences in average throughput obtained in the simulator remain fairly constant as the network size increases as depicted in Figure 6.13(b). This is largely a result of the influence of a traffic scenario that concentrates network traffic at the MPP and its immediate neighbouring mesh backbone nodes. The evaluated schemes are affected in differing ways by this traffic scenario. The Max. Power scheme benefits from shorter average path lengths and higher PDRs to the MPP but suffers from the increased levels of delay caused by greater MAC-level interference. PlainTC suffers the converse fate by having higher average path lengths and packet losses to the MPP but is compensated by accommodating more concurrent traffic flows due to better spatial re-use (see Section 6.3.6). However, PlainTC's low PDR negatively affects its average throughput in the portal-oriented scenario as losses are caused by the longer path length and the congestion caused by many competing traffic flows to the MPP.

PlainTC was found not to improve the throughput performance of the I-WMN and is not the only scheme to have been found to degrade the throughput. The CBTC scheme was reported in Gao et al. (2008) to degrade network throughput and the same work established throughput improvements that were produced for the MaxSR scheme. The CLUSTERPOW and COMPOW schemes reported in Kawadia and Kumar (2005) were also found to improve network throughput. The influence of node densities on throughput performance was revealed in Ramanathan and Rosales-Hain (2000). The CONNECT and BICONN schemes reported in Ramanathan and Rosales-Hain (2000) were found to improve network throughput at higher node densities and degrade throughput

TABLE 6.20: Average Throughput Achieved for Intra-mesh Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 328.17 | 327.54 |
| 9 | 263.91 | 263.85 |
| 10 | 227.43 | 221.50 |
| 11 | 213.87 | 211.09 |
| 12 | 196.51 | 180.82 |
| 13 | 192.63 | 184.25 |
| 14 | 185.11 | 170.92 |

TABLE 6.21: Average Throughput Achieved for Intra-mesh Traffic (Simulator)

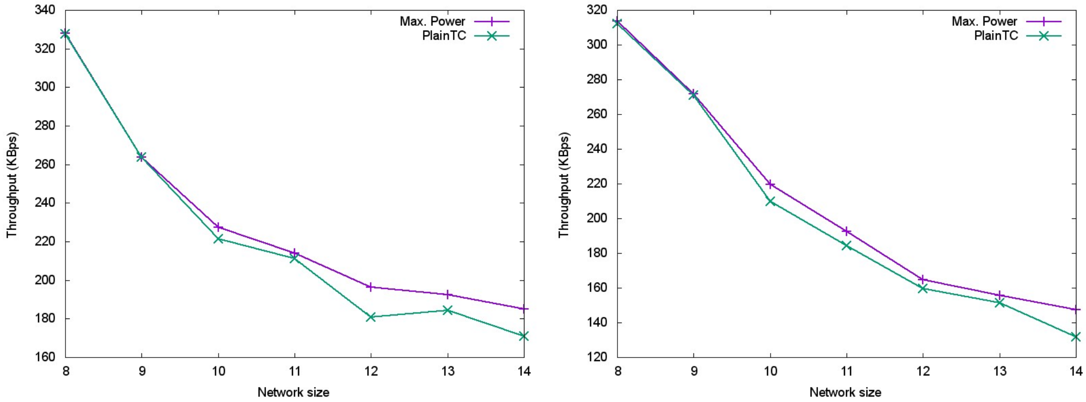
| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 586.02 | 500.92 |
| 40 | 565.17 | 463.97 |
| 60 | 488.19 | 397.73 |
| 80 | 410.08 | 374.57 |
| 100 | 376.50 | 321.88 |
| 120 | 313.16 | 269.39 |

TABLE 6.22: Average Throughput Achieved for Portal-oriented Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 8 | 313.57 | 312.27 |
| 9 | 271.82 | 270.83 |
| 10 | 219.67 | 209.71 |
| 11 | 192.72 | 184.18 |
| 12 | 164.57 | 159.73 |
| 13 | 155.79 | 151.52 |
| 14 | 147.35 | 131.61 |

TABLE 6.23: Average Throughput Achieved for Portal-oriented Traffic (Simulator)

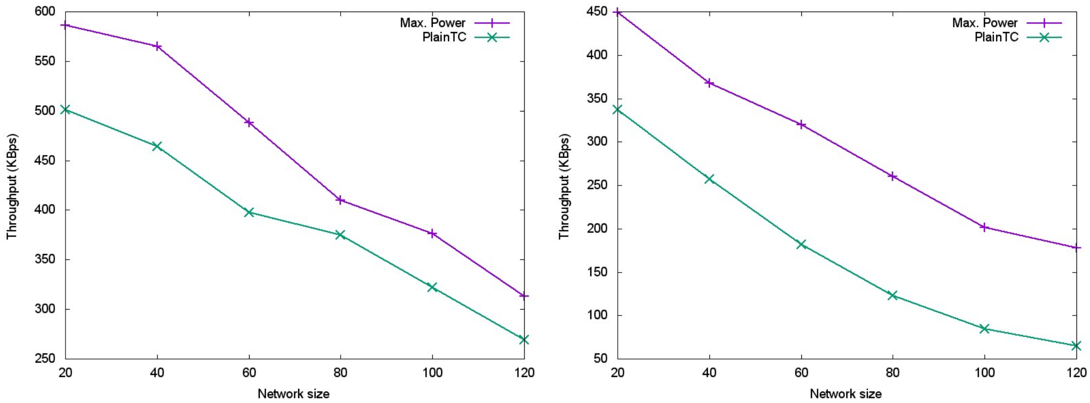
| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 449.37 | 437.06 |
| 40 | 367.67 | 357.30 |
| 60 | 319.85 | 281.81 |
| 80 | 260.27 | 223.02 |
| 100 | 201.45 | 184.42 |
| 120 | 178.23 | 165.05 |




(a) Intra-mesh traffic

(b) Portal-oriented Traffic

FIGURE 6.12: Throughput (Test-bed)




(a) Intra-mesh traffic

(b) Portal-oriented Traffic

FIGURE 6.13: Throughput (Simulation)

at the lower node densities. The influence of the interference range on throughput performance was shown in Aron et al. (2008a). The LM-SPT scheme with the minimum interference range was found to marginally underperform against Max. Power whilst the same scheme employing a larger interference range was found to significantly outperform the status quo. These prior results suggest that PlainTC's degradation of network throughput was to be expected as the scheme employed the minimum interference range[7] dictated by its CNN-based connectivity strategy and was not evaluated in a

[7]The transceiver powers used in the study reported in Aron et al. (2008a) were several factors higher than the minimum required for connectivity and would adversely affect the spatial re-use achieved in a network.

high-density network[8]. These factors have been shown to positively affect the network throughput and their absence in this evaluation negatively affected PlainTC.

### 6.3.8 Network Lifetime

Network lifetime is defined as the elapsed time until the first backbone node depletes its battery power supply. Thus, the network lifetime is defined as the duration for which all the backbone nodes exist on the network. The failure of a backbone node will cause all its associated client devices to be disconnected from the I-WMN. The network lifetime is therefore an important metric to consider for the battery-powered I-WMNs that are used in rural deployments. The purpose of this experiment is to determine whether the cumulative transceiver power savings produced by PlainTC will result in an extension of the network lifetime.

This subsection comprises only the results obtained from the network simulator as the test-bed nodes were not battery-powered at the time. Subsequent node lifetime experiments conducted on the test-bed are to be found in Oki (2013). The node lifetime results contained in Oki (2013) indicate that an approximation of PlainTC reduces the network lifetime. The first node dies after a duration of approximately 125000s (34.7hrs) if the battery power source is not replenished. Thus, despite the reduction in the node lifetime, the network lifetime may not be negatively affected if the battery is replenished within 34.7hrs. Battery-powered deployments are usually accompanied by renewable energy providers such as solar panels.

The simulation results contained in Table 6.24 indicate that PlainTC does not extend the network lifetime. Thus, the cumulative transceiver power savings produced by PlainTC for the various network sizes are insufficient to extend the duration until the first backbone node depletes its power supply. However, Table 6.24 also shows that the Max. Power scheme does not hold a significant advantage over PlainTC.

[8] This is not representative of the rural I-WMN deployments being considered.

TABLE 6.24: Network Lifetimes for the various backbone network sizes

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | 112.97 | 112.51 |
| 40 | 73.82 | 70.12 |
| 60 | 61.13 | 61.81 |
| 80 | 49.85 | 45.14 |
| 100 | 42.40 | 36.16 |
| 120 | 31.30 | 28.08 |

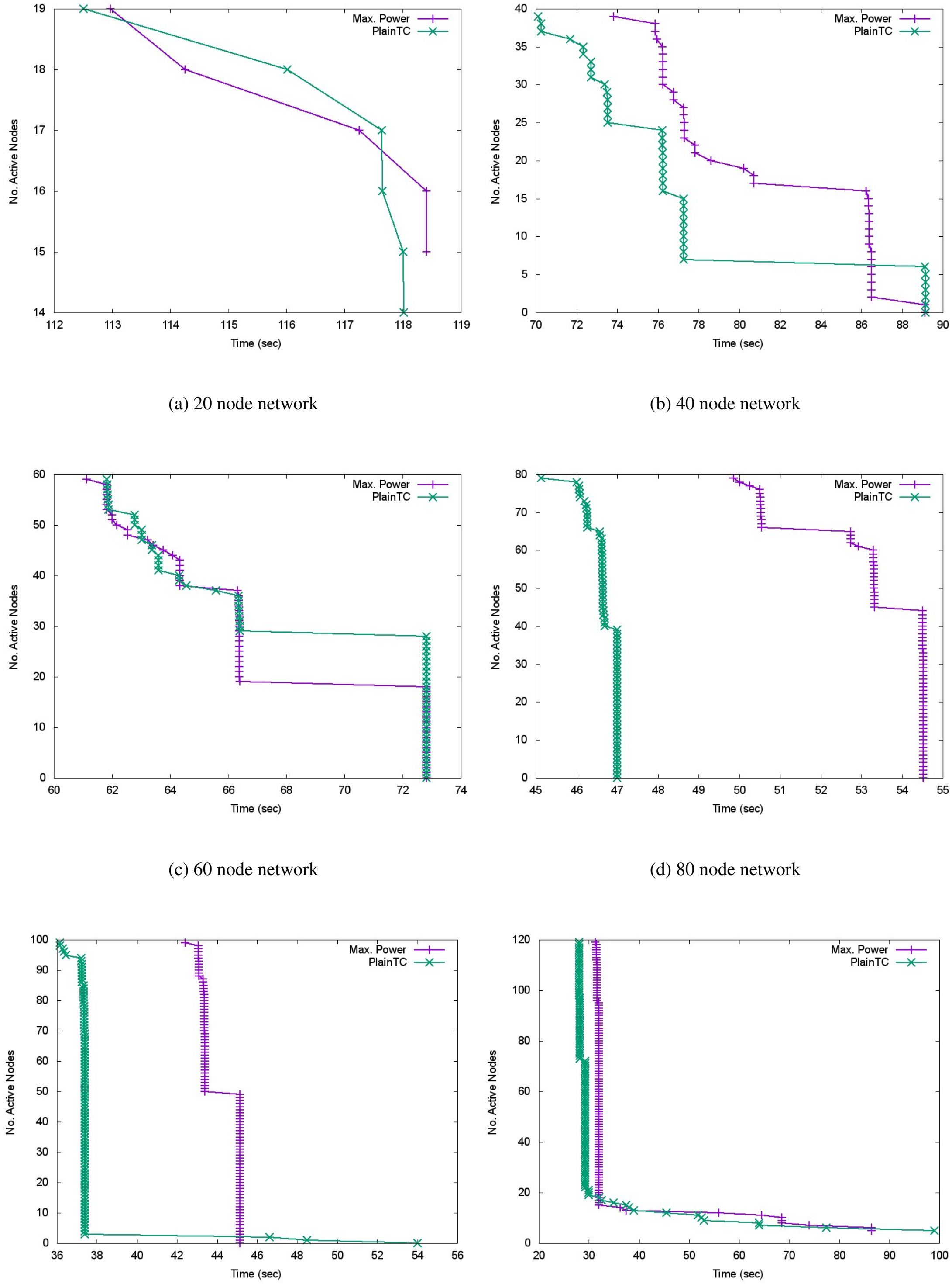
Other schemes found in the literature are shown to improve the network lifetime. The LMST and LM-SPT schemes found in Aron et al. (2008c) produced significant improvement in a 50-node network but these improvements were less-pronounced in a 100-node setting. The TAP scheme found in Hu et al. (2006) was found to match the network lifetime exhibited by the Max. Power scheme. The work reported in Liu et al. (2008) shows that the node density affects the network lifetime. Higher node densities were found to produce higher network lifetimes.

The reduction in the network lifetime attributable to PlainTC can also be explained by the effect that the reduction in node transceiver powers has on the resultant path lengths between source and destination nodes. Lower transceiver powers require data to traverse more hops to reach the intended destination. More intermediate nodes are required to expend energy during packet reception, processing and re-transmission. This negates the transceiver power savings achieved by the individual intermediate nodes.

The Max. Power scheme is disadvantaged by its use of the maximum transceiver power at every network node, but benefits from shorter path lengths to the destination nodes. Fewer intermediate nodes are required to expend energy during packet reception, processing and re-transmission.

The node failure profile for both schemes over the various network sizes in depicted in Figure 6.14. This figure depicts the propensity for several nodes to fail almost simultaneously. This suggests that collections of nodes experience similar battery-depletion rates. A deeper inspection of this phenomenon results in Figures 6.15 and 6.16.

Figures 6.15 and 6.16 depict the relationship between the lifetimes experienced by the various network nodes and their positions in the network relative to an imaginary




(a) 20 node network

(b) 40 node network

(c) 60 node network

(d) 80 node network

(e) 100 node network

(f) 120 node network

FIGURE 6.14: Network Lifetime (Simulation)

central point. As the network size increases, bands of nodes with similar lifetimes but with differing distances away from the network centre can be observed. This can be interpreted as the existence of a routing highway, comprising high-quality links[9] across the backbone network, along which the bulk of the network traffic is transported.

The existence of the routing highways is further supported by the presence of subsequent secondary and, in some instances, tertiary bands of nodes that share similar node lifetimes. This suggests that as the nodes comprising the current routing highway deplete their batteries, the routing protocol creates an alternate highway with a differing set of nodes spread across the backbone network. This process repeats until there are insufficient nodes still alive to facilitate routing. Figures 6.15 and 6.16 show that the time between the failures of successive highways shortens as the network size increases. This occurs because the secondary and tertiary nodes are still active participants in the network as they may be sources or destinations of traffic flows. In addition, these secondary and tertiary highway nodes may neighbour a current primary highway node and are thus expending energy in receiving and decoding packets meant for the neighbouring node that is currently participating in the routing highway.

The existence of the routing highway does not mean that network traffic is routed over the entire length of the highway but rather that the majority of the network traffic flows travel over a subset of the highway en-route to the intended destination. The existence of collections of nodes that deplete their batteries at similar rates but are spread across the network means that the routing highways consist of nodes with varying transceiver power outputs[10] but similar total energy expenditures. This means that the contributions of the transceiver power outputs and traffic forwarding functions to the total energy expenditure of the node will differ across the highway nodes.

Nodes furthest away from the network centre amongst the highway nodes experience a greater contribution from their transceiver power output towards their battery depletion. Nodes that are closer to the network centre experience a greater contribution from the energy spent during traffic forwarding to their battery depletion.

[9]Due to preference of the OLSR+ETX combination for stable, high-quality links.
[10]Refer to Section 6.3.3 for the node's transceiver power assignment patterns.

The results for the largest network size considered (120 backbone nodes) are depicted in Figure 6.16. This large network size allows for the outlying nodes to be relatively uninvolved in forwarding network traffic. The battery depletion of these outlying nodes is dominated by their high transceiver power outputs. Thus, the outlying nodes subjected to the Max. Power scheme depicted in Figure 6.16(e) deplete their batteries earlier than their PlainTC counterparts depicted in Figure 6.16(f).

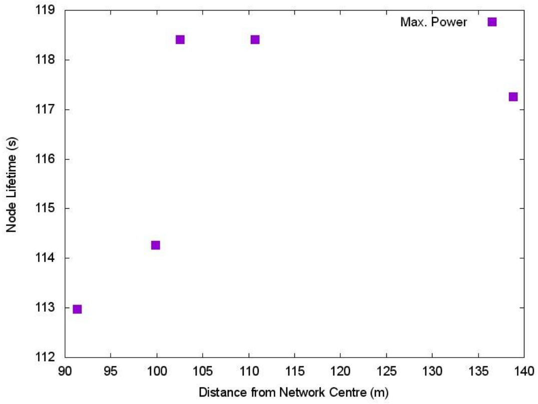
### 6.3.9 Energy Cost of Data Transfer

The energy cost of data transfer is an important measure of the efficacy of a TC scheme. However, together with the network lifetime metric, the energy cost metric has been lacking in the TC literature. The energy cost metric is a measure of energy efficiency since an inverse relationship between the energy cost in Joules and efficiency exists.

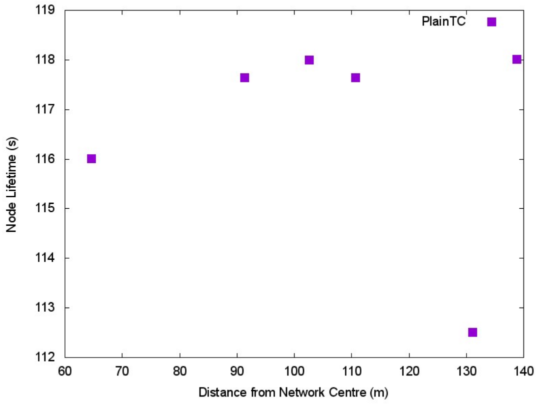
The energy cost values are contained in Table 6.25. PlainTC is found to be less energy efficient than the Max. Power scheme and the underperformance is shown to grow with increases in the network size. This is a result of the longer paths to destinations as the values contained in Table 6.25 represent the average energy cost over the path taken to deliver a KB of data. Thus, the higher path lengths created by PlainTC will incur higher cumulative transmission and processing energy costs. The earlier network lifetime experiment showed that the processing energy cost is greater than the transmission energy cost for PlainTC[11]. Thus, PlainTC reduces the transmission energy cost for a path to a destination but increases the path's processing energy costs.

Other schemes are reported to be more energy-efficient than the Max. Power status quo. Both the TAP scheme found in Hu et al. (2006) and the FLSS scheme reported in Li and Hou (2006) transmit more data per joule than the Max. Power scheme. PlainTC's combination of longer path lengths, higher node degree and shorter network lifetime combine to reduce the energy cost of data transfer.

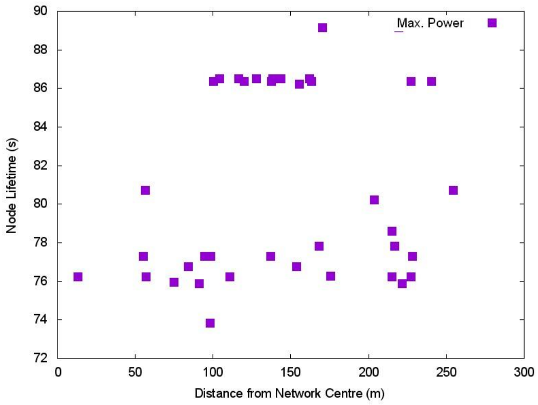
[11]This is also alluded to in an earlier finding reported in Santi (2005) that the wireless transceiver accounts for between 15% to 35% of the total power consumed by a wireless node.



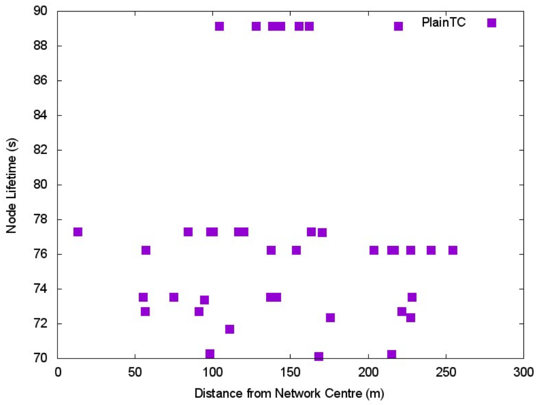

(a) 20 node network



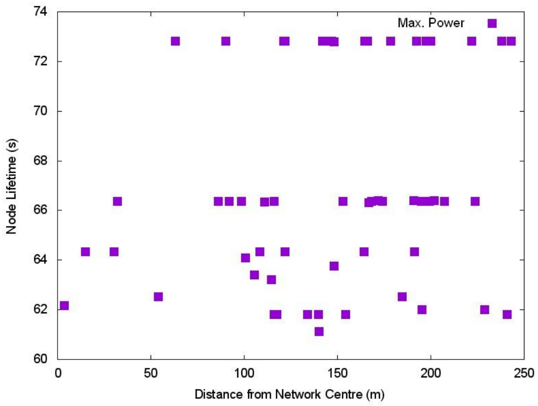

(b) 20 node network



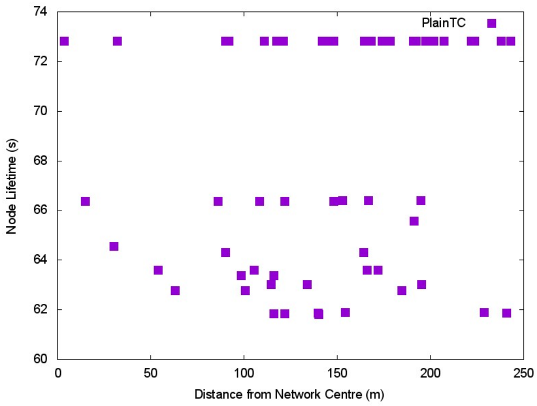

(c) 40 node network




(d) 40 node network

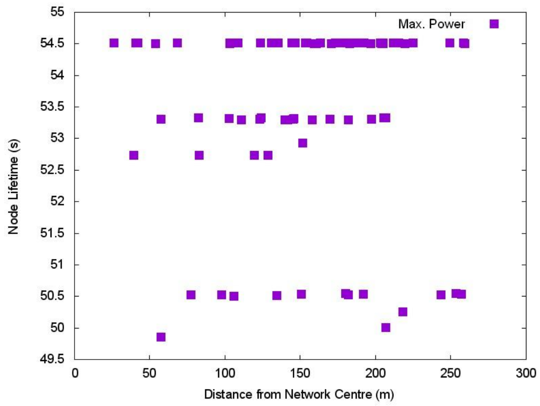



(e) 60 node network

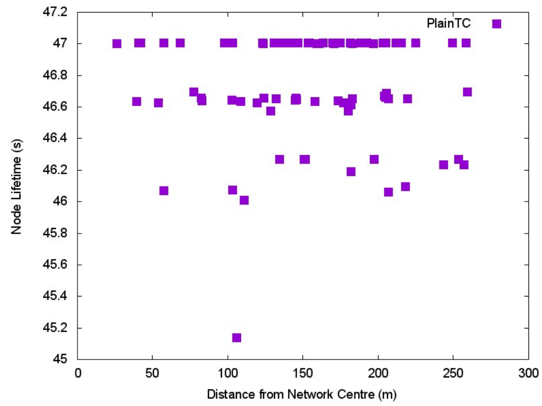



(f) 60 node network

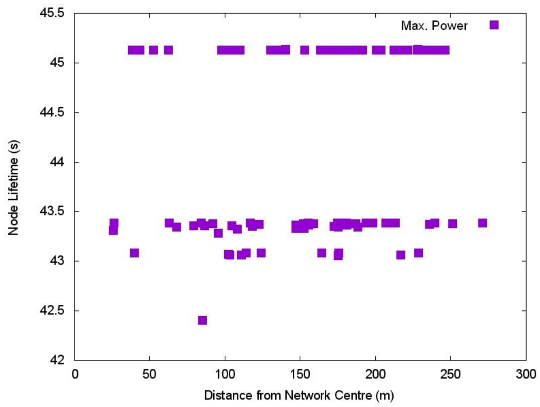
FIGURE 6.15: Influence of Distance from Network Center on Node Lifetimes



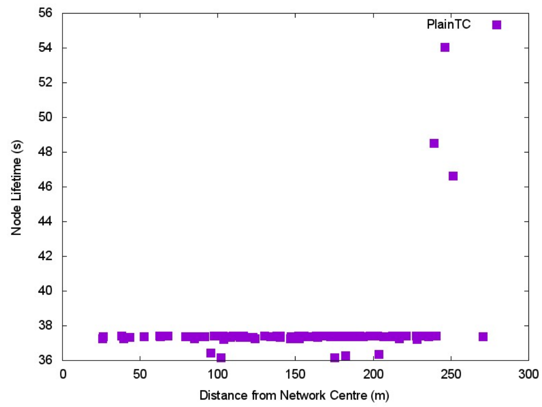

(a) 80 node network



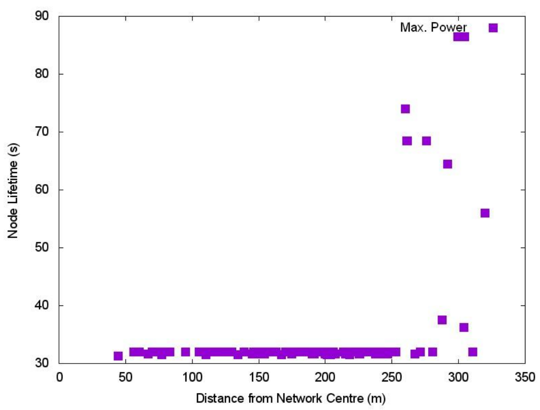

(b) 80 node network



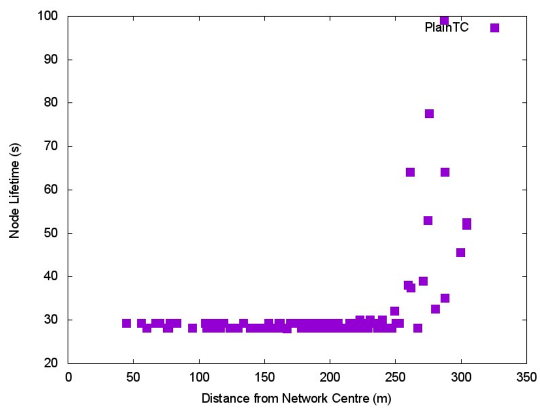

(c) 100 node network




(d) 100 node network




(e) 120 node network




(f) 120 node network

FIGURE 6.16: Influence of Distance from Network Center on Node Lifetimes (continued)

TABLE 6.25: Average Energy Expended per KB Transmitted (Simulation)

| Network Size | Max. Power | PlainTC |
|---|---|---|
| 20 | $3.02*10^{-3}$J | $3.55*10^{-3}$J |
| 40 | $9.56*10^{-3}$J | $1.23*10^{-2}$J |
| 60 | $2.01*10^{-2}$J | $2.44*10^{-2}$J |
| 80 | $3.91*10^{-2}$J | $4.73*10^{-2}$J |
| 100 | $6.42*10^{-2}$J | $8.59*10^{-2}$J |
| 120 | $1.22*10^{-1}$J | $1.59*10^{-1}$J |

## 6.4 Chapter Summary

This Chapter has presented the evaluation of PlainTC on both the test-bed and simulation platforms. The evaluation was conducted to assess the viability of PlainTC for deployed I-WMNs. PlainTC's dynamic, run-time behaviour as well as its effect on a backbone network's energy savings and data traffic performance were investigated.

The test-bed platform allowed for the dynamic, real-time behaviour of PlainTC to be observed from both the network and node perspective. This provided a unique perspective on the operation of a TC scheme. These observations highlighted the effect of the iterative, gradual transceiver power adjustment process on the maintenance of network connectivity and the required neighbourhood size. The network view was found to hide the neighbourhood and link quality instabilities caused by PlainTC.

The traditional evaluations indicate that PlainTC is effective at maintaining network connectivity, producing cumulative transceiver power savings, reducing MAC-level interference and maintaining high PDR levels. These results corroborate existing results in the literature and extend the body of knowledge as PlainTC, although materially different from the existing TC schemes, was still able to achieve similar results.

PlainTC's throughput, network lifetime and energy-efficiency results differ from those found in the literature. The throughput result is influenced by the rural I-WMN deployment scenario being considered where a low node density is prevalent and interference ranges are not artificially increased to improve network performance. The network lifetime and energy-efficiency results presented in this chapter differ from those in the literature as the energy consumption model used in this study is based on an actual I-WMN node.

The traditional evaluation has also presented unique experiments for which no previous data can be found in the TC literature. Centrally located nodes were found to be more likely to be the source of the cumulative transceiver power savings measured. The study also documented the increase in the number of concurrent transmissions that can be supported by a network backbone employing a TC scheme. In addition, the existence of primary, secondary (and even tertiary) routing highways wherein the participating backbone nodes exhibit similar energy-depletion rates and failure times was shown.

The simulation setup closely matched that of the test-bed and no inconsistent results were reported when using these two evaluation platforms. An objective view of the entire set of results indicates that PlainTC impairs network performance and energy efficiency but improves the spatial re-use of the shared communication medium. The instabilities caused at the individual nodes (refer to Section 6.2.5) does not recommend the use of PlainTC in its current form within a deployed I-WMN. It can be reasoned that PlainTC's network underperformance may be exacerbated by the topology instability caused by constant transceiver power output level and the resultant neighbourhood size changes that the scheme causes. However, a mechanism that could better regulate the adjustment of node transceiver power output levels may help to reduce the topology instability. A reduction in the topology instability should produce improvements in data traffic performance that can be detected by the metrics being employed.

The next chapter (Chapter 7) explores a mechanism for mitigating the topology instability caused by the constant changes to a node's transceiver power output levels and resultant neighbourhood sizes. The mechanism employs contextual information to augment the use of the CNN to determine whether a transceiver power change should occur.

# Chapter 7

# Using Context to reduce Topology Instability

## 7.1 Introduction

The previous Chapter assessed the viability of PlainTC as a means of improving the energy-efficiency and performance of an I-WMN. The test-bed- and simulation-based evaluations indicate that PlainTC is not currently capable of improving energy-efficiency and performance levels. Observations of PlainTC's effect on the behaviour of test-bed nodes suggest that the nodes adjust their transceiver power output levels too frequently. The regular changing of nodes' transceiver power output levels was shown to have a multiplier effect on the resultant number of changes observed for the neighbourhood size, network size and link quality variables. The frequent changes to these four variables have been shown to lead to topology instability.

Topology instability is detrimental to the QoS levels offered by the backbone network. This chapter presents a mechanism for reducing the topology instability caused by PlainTC. The mechanism employs contextual information collected by the node's firmware and the OLSR routing protocol. The contextual information is meant to reduce the number of transceiver power changes executed by a node. A reduction in the number of changes to the transceiver power output should cause reductions in the other three parameters as well. Reduced topology instability should be expected.

There are several technical challenges with our intended use of contextual information to address topology instability in practical settings. Firstly, the required contextual information should be made available by the node firmware. This prevents having to change the firmware source code and run the risk of introducing unintended bugs into the code via a lack of adequate testing. Secondly, attention should be paid to the timing of the data collection. Infrequent data collection will result in data that quickly becomes incomplete and outdated. This leads to inaccurate decision-making. Data collection that is too frequent can lead to data duplication which does not improve the accuracy of decision-making. Thirdly, the speed of data processing and decision-making must be considered. If the decision-making process is too complex and takes too long to complete then the decision becomes irrelevant as it does not react to the present network conditions. Thus, a simple method is preferred over a computational complex one as network conditions are subjected to continuous changes that are exacerbated by a TC scheme. In addition, any improvement to PlainTC that employs contextual information is subject to the original operational scope and design criteria contained in Sections and respectively.

Section 7.2 describes a method for quantifying the changes to a node's context. The quantum of context change is meant to be compared to a context-change threshold to determine whether an adjustment to a node's transceiver power output level should occur. An enhanced version of PlainTC called PlainTC+ is introduced in Section 7.3. PlainTC+ uses the results of the method to quantify context change and the context-change threshold in an attempt to reduce topology instability. Section 7.4 contains the subsequent test-bed-based evaluation of PlainTC+.

The usage of the Principal Component Analysis statistical method in Section 7.2 for the purpose of quantifying a context variable's contribution to topology instability extends current knowledge. Kubo et al. (2010) used context information to improve routing performance whilst Matos et al. (2012) used context information to improve the QoS offered to client devices. In contrast, this study uses context information to reduce

TABLE 7.1: Total number of changes in node parameters over a 24-hour period

| No. changes in Transceiver Power | No. changes in Neighbourhood Size | No. changes in Network Size | No. changes in Link Quality |
|---|---|---|---|
| 1683 | 2515 | 5300 | 7433 |

topology instability caused by TC. The reduction in topology instability is demonstrated in Section 7.4. Existing studies found in Krebs et al. (2010) and Aziz et al. (2011) have addressed topology instability in WMNs in differing ways. Krebs et al. (2010) proposed clustering as a remedy for topology instability whilst Aziz et al. (2011) proposed a flow control mechanism to smooth data traffic. In contrast, this study reduces topology instability by applying a context-change threshold to regulate a node's transceiver power adjustment.

## 7.2 Reducing Topology Instability

Section 6.2.5 in the previous chapter exposed the topology instability caused by PlainTC. Topology instability cannot be measured directly but it can be inferred from the number of recorded changes to the transceiver power output level, neighbourhood size, network size and link quality variables respectively for the entire network. These variables are chosen for observation as they are easily collected during the normal operation of a test-bed node. Thus, the collection of the values associated with these variables does not place additional demands on resource-constrained I-WMN nodes. The number of changes observed for each of the four variables over a twenty-four hour period is contained in Table 7.1. The detailed breakdown over twenty-four one-hour periods was shown in Section 6.2.5.

The four variables are taken to represent the context of a node as they represent a node's view of its local neighbourhood and the global network. A change to any of these four variables represents a change to the node's context. The purpose of this section is two-fold. The first objective is to provide a process that can be followed to determine each variable's contribution to the observed change in context. The contribution associated with each variable is a measure of the relative impact of a change in that variable on the overall node context. The impact can be quantified as a weight associated with each of the four observed variables. The second objective is to determine the appropriate threshold value for changes in context which will reduce the number of transceiver power output changes and the resultant instability[1].

Changes in the observed variables are referred to as events. The weights associated with each of the observed variables are meant to identify the high-impact events that PlainTC should respond to. This differs from PlainTC's current operation where events taking place outside of a node's immediate vicinity cause the node to adjust its transceiver power output. The envisaged result is a reduction in the number of changes made to a node's transceiver power levels and a reduction in the multiplier effect that such changes have on the other observed variables[2].

The weights associated with each variable are meant to be used to quantify the change in context experienced by each test-bed node. The process that is established in this section can be used on any deployed I-WMN to determine the appropriate weightings for that particular deployment. The use of this process will be demonstrated using data obtained from the indoor test-bed introduced in Chapter 5.

The data contained in Table 6.1 is subjected to a statistical method described in Jolliffe (2002) called Principal Component Analysis (PCA). PCA is commonly used to reduce the dimensionality of large data sets but it can also be used to determine the contribution of each variable to the total variability contained in multivariate data. PCA is being employed, in this instance, to determine the contribution (weight) of each variable to the overall context change experienced by a node.

The *FactoMineR* package introduced in Le et al. (2008) is used within the *R* statistical environment to perform the PCA. Apart from being able to perform classical PCA, *FactoMineR* also has a mechanism for determining the contribution of a variable to total variability of multivariate data. This mechanism is documented in tools for high-throughput data analysis (2015). This makes *FactoMineR* and its PCA methodology a

[1]Refer to Section 6.2.5 for a discussion of the instabilities caused by frequent transceiver power output changes.

[2]Refer to Section 6.2.5.

reliable choice for determining the contribution of each of the afore-mentioned variables to the variability in topology instability.

The process defined by *FactoMineR* for calculating the variable weights is as follows and the associated sequence of *R* commands can be found in Appendix IV.

Step 1

The first step is to load the contents of Table 6.1 into *R*.

Step 2

The second step is to perform the PCA on the input data.

Step 3

The third step determines the proportion of variation retained by the Principal Components. The amount of variability retained by each Principal Component is calculated from the *eigenvalues*. The variability for each component is depicted below:

```
       eigenvalue   percentage of variance   cumulative
comp 1 3.35336277               83.8340693     83.83407
comp 2 0.43964847               10.9912116     94.82528
comp 3 0.18722414                4.6806035     99.50588
comp 4 0.01976462                0.4941155    100.00000
```

Eigenvalues less than 1 are commonly used to eliminate their associated dimensions as these dimensions do not greatly account for the total variance in the data. Thus, only the first component is selected for further use as it has an eigenvalue greater than 1 and accounts for approximately 84% of the total variance contained in the data. Contributions towards the total variance contained in the data are depicted in Figure 7.1.

Step 4

The fourth step determines the percentage contribution of each variable on the total variance explained by the identified components:

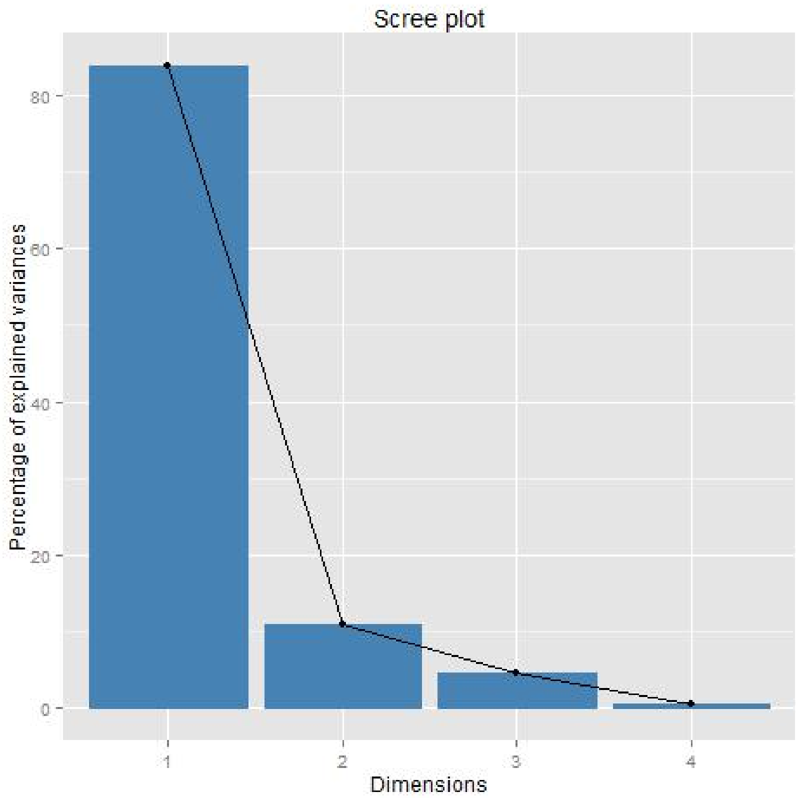



FIGURE 7.1: Contribution of dimensions towards the total variability in the observed data

| | Dim.1 |
|---|---|
| Transceiver.Power | 27.95265 |
| Network.Size | 24.82228 |
| Neighbourhood.size | 27.66850 |
| Link.Quality | 19.55657 |

These contributions correspond to the weights that can be associated with each variable and represent the relative impact of a change in that variable on the context-change experienced by a node. This step represents the end of the PCA on the observed data.

Step 5

The fifth step uses the normalised weights (obtained in the previous step) to determine the appropriate context-change threshold for the network. The goal of the threshold value is to reduce the number of changes made to the transceiver power output level of a network node. Thus, the threshold acts as a sentinel value that determines whether an observed change is sufficiently significant to warrant an adjustment to a node's transceiver power output level.

### 7.2.1 Formulating the Threshold Value

The weightings derived for each the four observed variables and knowledge of the multiplier effect of the transceiver power variable on the other three variables provide valuable input into the formulation of the context-change threshold value. This value has to be sufficiently high to reduce the number of changes to the transceiver power output variable but sufficiently low to cause PlainTC to react to important events.

An important event is defined as one that takes place within the immediate neighbourhood of an affected node and affects a variable with network-wide impact such as the network size. Less important events originate from outside of the immediate neighbourhood. These definitions allow for a node to react to important events originating in its immediate neighbourhood by adjusting its transceiver power output whilst maintaining the current transceiver power output for events originating elsewhere. A balance between reducing the number of transceiver power changes and reacting to important events can thus be achieved.

The appropriate threshold value can be determined through experimentation with the test-bed. This would entail varying the threshold value and analysing the effect of the various threshold values on reducing the number of observed transceiver power output changes and the responsiveness of the TC scheme. This approach is eschewed in favour of reasoning.

Threshold values of 0.1956, 0.2482, 0.2767 and 0.2795 respectively, are rejected as these hurdles are too low to ensure that the TC scheme only reacts to local events with network-wide impact. A value of 1.0000 is too high and the hurdle would only be met in extreme circumstances such as network partitioning or network merging immediately after the affected node has adjusted its transceiver power output. The TC scheme would thus be rendered unresponsive to all events, even those in its immediate neighbourhood, under normal circumstances.

The only means that a TC scheme possesses in order to react to network events is to adjust the transceiver power output of the affected node. Calculating the threshold using the transceiver power variable with any combination of the other variables would enable continuous adjustment of node transceiver power output levels. This is a situation to be avoided as the aim of the threshold value is to reduce the number of transceiver power changes performed by the network nodes.

PlainTC currently adjusts its transceiver power output in order to maintain the desired CNN at each network node. This CNN value is derived from the network size, and the network size variable has been shown to change quite frequently. Thus, the CNN value changes frequently in response to the network size variable and the resultant transceiver power adjustment is attempted. The transceiver power change should only occur if the change in the observed network size is caused by a change in the observed neighbourhood size. This would force the TC scheme to react only to events in the affected node's immediate neighbourhood which have network-wide impact. Thus, the network size and neighbourhood size variables must contribute to the threshold value. The link quality variable changes most frequently because it encapsulates natural variations in link quality as well as changes caused by PlainTC+. Thus, the link quality variable must also contribute to the threshold value. Therefore, the most appropriate threshold value to force the TC scheme to react only to local events with a network-wide impact is one that ensures that PlainTC+ will only adjust a node's transceiver power output in situations where the affected node simultaneously experiences changes in neighbourhood size, network size and in link quality. The selected threshold value is derived from the weightings determined in Step 4 and is computed as:

$$\begin{aligned} contextChangeThreshold &= (0.2767 \times 1) + (0.2482 \times 1) + (0.1956 \times 1) \qquad (7.1) \\ &= 0.2767 + 0.2482 + 0.1956 \\ &= 0.7205 \end{aligned}$$

## 7.3 PlainTC+

PlainTC+ is an amalgamation of the original PlainTC algorithm and the results of the process to reduce topology instability. The process (contained in the previous Section) produces two outcomes: the first outcome is the determination of the weights associated with each observed variable and the second outcome is the appropriate context-change threshold value for the observed network. The weights associated with each observed variable are the normalised contribution of each variable to the total variability in the collected data. These contributions (weightings) are determined in Step 4 in the process described in the previous Section.

The context-change for the observed test-bed network can be computed as:

$$contextChange = (0.2795 \times A) + (0.2767 \times B) + (0.2482 \times C) + (0.1956 \times D) \quad (7.2)$$

where:

$$A = \begin{cases} 1, & \text{if } |\Delta transceiverPower| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7.3)$$

$$B = \begin{cases} 1, & \text{if } |\Delta neighbourhoodSize| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7.4)$$

$$C = \begin{cases} 1, & \text{if } |\Delta networkSize| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7.5)$$

$$D = \begin{cases} 1, & \text{if } |\Delta linkQuality| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7.6)$$

The result is a normalised value in [0,1] that is independently computed by each node, with the result depending upon the variable changes being reported by the node. This quantified context-change value can now be subjected to the appropriate threshold value within PlainTC+ in order to reduce the number of transceiver power output changes performed by the network nodes.

The resultant algorithm employed by PlainTC+ is shown in Algorithm 2. The use of the CNN value is augmented by the context-change threshold value when deciding to adjust a node's transceiver power output. The potential reduction of a node's transceiver power output is a key motivator for the use of a TC scheme in the I-WMN. Thus, PlainTC+ is designed to allow for the easy reduction of a node's transceiver power output by basing the adjustment decision only upon the CNN to be maintained. Raising a node's transceiver power output requires that the quantified change being observed meets or exceeds the context-change threshold value. The threshold value is defined in line 8 and the additional context-change condition can be found in line 16 of Algorithm 2. These additions to the algorithm are meant to reduce the number of transceiver power adjustments made by a node whilst allowing for savings in a node's transceiver power output to be achieved.

Despite the addition of the context-change threshold mechanism to PlainTC+, the computational complexity is not adversely affected. In addition, the requirements of a TC scheme discussed in Section 7.1 are still met. Thus, PlainTC+ is a practical TC scheme that can be employed on low-cost, resource-constrained I-WMN nodes.

The associated source code of PlainTC+ for the Linksys WRT54GL node can be found in Appendix IV.

## 7.4 Evaluation of PlainTC+

PlainTC+ is designed to reduce the topology instability caused by transceiver power changes. This is achieved by limiting the number of power changes performed by a node in the backbone network. PlainTC+ is evaluated against PlainTC to determine whether the context-change mechanism can successfully reduce topology instability. This section only contains test-bed-based experiments as only the test-bed allowed for

**Algorithm 2** Execution Logic of PlainTC+

```
no_nodes ← getRoutingTableSize()
current_no_neighbours ← getNeighbourhoodSize()
CNN ← ⌈5.1774 log(no_nodes)⌉
max_tx_power ← 78qdBm
min_tx_power ← 1qdBm
current_tx_power ← getCurrentTxPower()
tx_power_change_level ← 3qdBm
context_change_threshold ← 0.7205
if current_tx_power ≤ max_tx_power then
    if current_no_neighbours > CNN then
        if (current_tx_power − tx_power_change_level) ≥ min_tx_power then
            current_tx_power ← current_tx_power − tx_power_change_level
        end if
    end if
    if current_no_neighbours < CNN then
        if getContextChange() ≥ context_change_threshold then
            if (current_tx_power + tx_power_change_level) ≤ max_tx_power then
                current_tx_power ← current_tx_power + tx_power_change_level
            end if
        end if
    end if
end if
```

the observation of topology instability. The methodology employed for each experiment remains unchanged from that used in the previous chapter (and described in Chapter 5).

### 7.4.1 Topology Instability

The number of changes to the observed network variables when using PlainTC+ are listed in Table 7.2. PlainTC+ is found to significantly reduce the number of changes for each observed variable[3]. The context-change threshold mechanism produced a 45% reduction in the number of transceiver power changes recorded over a 24-hour period. This result was produced by using the threshold value to suppress unnecessary transceiver power increases and compares favourably with a similar threshold mechanism employed to reduce route flapping. This *dampening threshold* mechanism was found to produce a 60% reduction in routing changes in Ramachandran et al. (2007). It

[3]Refer to Section 6.2.5 for PlainTC's instability performance.

must be remembered that PlainTC+ only applied its context-change threshold mechanism to transceiver power increases. It is expected that PlainTC+ would have achieved a larger reduction in instability if the threshold had been applied to transceiver power decreases as well. However, in the context of TC, such a move would have been at the expense of cumulative transceiver power savings.

The previous chapter established the de-stabilising effect that changing a node's transceiver power output has on its neighbouring nodes. Thus, the reduction in the number of transceiver power output changes also caused reductions in the other variables observed within the network. Reductions of 36%, 37% and 38% were recorded for the neighbourhood size, network size and link quality variables, respectively. These values represent a significant improvement to the stability of the network topology.

### 7.4.2 Additional Experiments

The effect of the context-change threshold on both the maintenance and the performance of the network is considered.

#### 7.4.2.1 Effect on the OLSR routing protocol

The OLSR routing protocol reports both the number of neighbouring nodes and the link quality associated with each neighbour. These metrics are affected by transceiver power output changes and the effect of a reduction in the number of transceiver power changes is depicted in Figures 7.2 and 7.3.

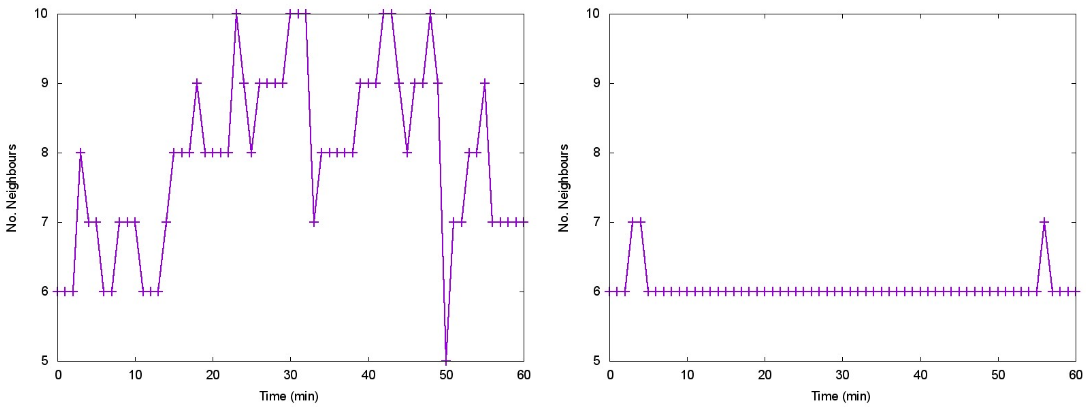
The restriction on transceiver power changes imposed by the context-change threshold helps to maintain a stable one-hop neighbourhood, as depicted in Figure 7.2. PlainTC+ causes a node and its neighbours not to deviate from the required CNN. This results in a stable neighbourhood size. Nodes may reduce their transceiver power output level to maintain the required CNN but the context-change threshold causes the node to avoid increasing its transceiver power output to maintain a new CNN when a new node joins

TABLE 7.2: Number of changes in node variables over a 24-hour period

| Time Period | No. changes in Transceiver Power | No. changes in Neighbourhood Size | No. changes in Network Size | No. changes in Link Quality |
|---|---|---|---|---|
| 08:00-08:59 | 59 | 100 | 198 | 277 |
| 09:00-09:59 | 40 | 68 | 143 | 199 |
| 10:00-10:59 | 39 | 66 | 139 | 195 |
| 11:00-11:59 | 39 | 70 | 146 | 202 |
| 12:00-12:59 | 38 | 65 | 139 | 196 |
| 13:00-13:59 | 37 | 68 | 136 | 189 |
| 14:00-14:59 | 36 | 61 | 128 | 184 |
| 15:00-15:59 | 35 | 60 | 124 | 173 |
| 16:00-16:59 | 38 | 68 | 151 | 211 |
| 17:00-17:59 | 37 | 68 | 147 | 201 |
| 18:00-18:59 | 39 | 66 | 119 | 166 |
| 19:00-19:59 | 38 | 71 | 135 | 189 |
| 20:00-20:59 | 38 | 65 | 137 | 192 |
| 21:00-21:59 | 36 | 60 | 131 | 183 |
| 22:00-22:59 | 37 | 65 | 127 | 178 |
| 23:00-23:59 | 42 | 70 | 152 | 207 |
| 00:00-00:59 | 36 | 59 | 125 | 171 |
| 01:00-01:59 | 37 | 67 | 136 | 190 |
| 02:00-02:59 | 35 | 64 | 135 | 186 |
| 03:00-03:59 | 36 | 66 | 140 | 196 |
| 04:00-04:59 | 37 | 67 | 129 | 179 |
| 05:00-05:59 | 40 | 69 | 148 | 199 |
| 06:00-06:59 | 37 | 62 | 128 | 184 |
| 07:00-07:59 | 36 | 65 | 139 | 196 |
| | | | | |
| **Total** | **922** | **1610** | **3332** | **4643** |

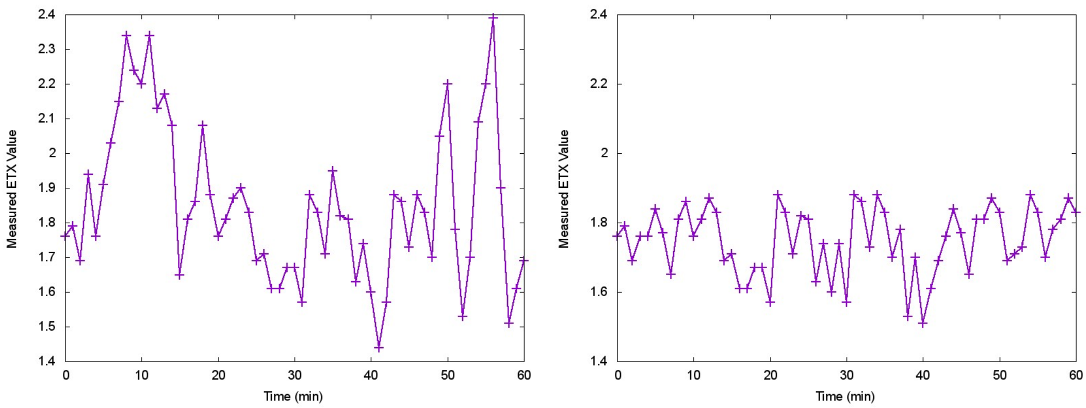
the network and this new node is not an immediate neighbour. Any increases in neighbourhood size are either caused by neighbouring nodes in the process of adjusting their own transceiver power outputs to maintain their CNN or by the natural variations in link quality which cause temporary loss of links to neighbours.

The improved network stability offered by PlainTC+ is also found to reduce the severity of the fluctuations in observed link quality. This is depicted in Figure 7.3. Nodes do not adjust their transceiver power output levels as frequently as with PlainTC. This improves the performance of the probe packets sent out by OLSR to calculate the ETX metric and leads to reduced route flapping caused by variations in the route cost metric.




(a) PlainTC (b) PlainTC+

FIGURE 7.2: Observed Neighbourhood Size




(a) PlainTC (b) PlainTC+

FIGURE 7.3: Observed Link Quality

#### 7.4.2.2 Transceiver Power Output

PlainTC+ results in a lower cumulative transceiver power output level than PlainTC (see Figure 7.4). The power savings are a result of PlainTC+ subjecting any increases in transceiver power output to the context-change threshold. This prevents nodes from increasing their transceiver power outputs in response to changes in network size that arise from outside of their immediate neighbourhood. The new CNN to be maintained in such scenarios is over-ruled by the context-change threshold. Therefore, nodes within an unaffected neighbourhood do not adjust their transceiver power output levels upward

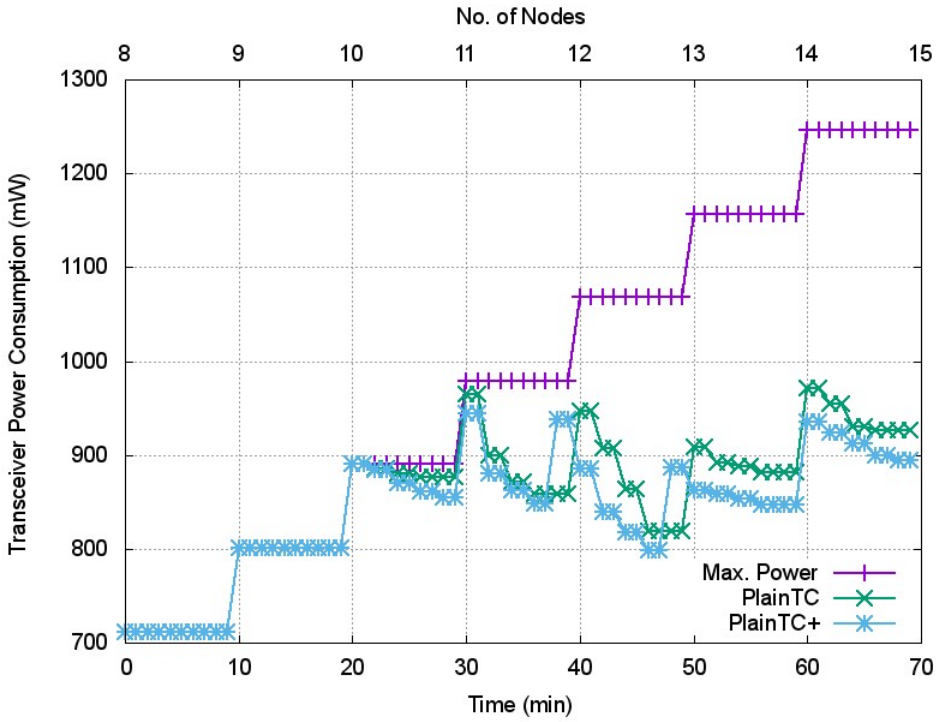



FIGURE 7.4: Dynamic Adjustment of Transceiver Power by PlainTC+

with PlainTC+ whereas these same nodes increase their transceiver power levels when employing PlainTC.

#### 7.4.2.3 Network Connectivity

Figure 7.5 shows that PlainTC+ reduces the time to establish network connectivity when changes to the network size occur. This reduction (when compared to PlainTC) is caused by the positive effect of the context-change threshold on the workings of the OLSR routing protocol.

Network connectivity is determined by the availability of routes to all possible destinations. The reduction in network instability caused by the context-change threshold aids in the propagation of routing information by OLSR. The routing updates propagate faster due to the increased stability of links between neighbouring nodes. Thus, the routing tables at each network node report routes for every other node at a faster rate than with PlainTC.

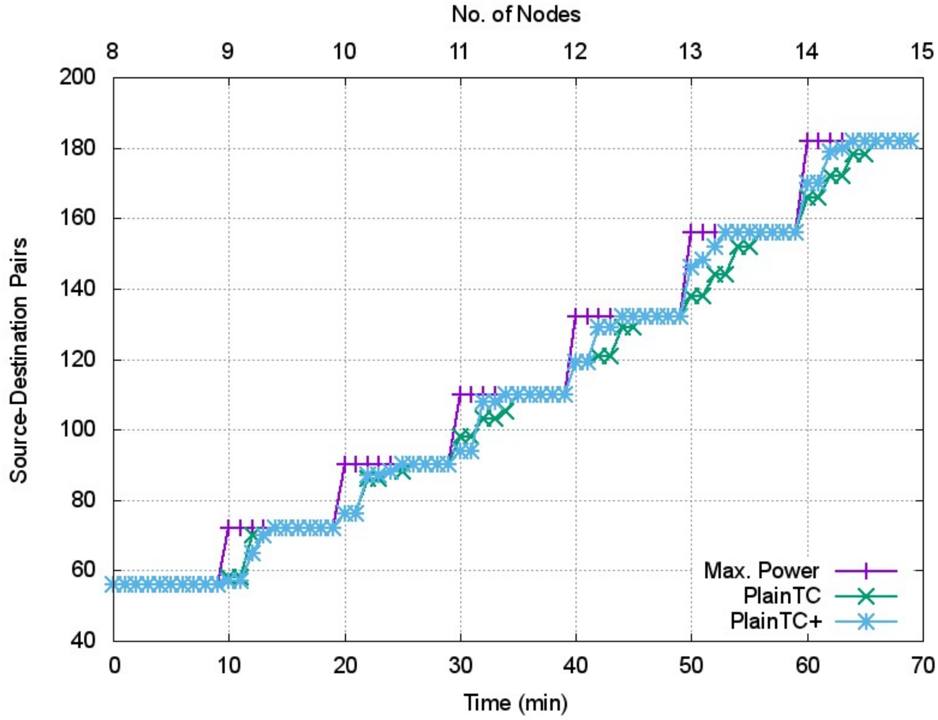



FIGURE 7.5: Dynamic Establishment of Network Connectivity by PlainTC+

#### 7.4.2.4 PDR & Throughput

PlainTC+ outperforms PlainTC in delivering data traffic to the intended destinations. The improvements in topology stability offered by PlainTC+ result in PDR and throughput increases for both the Intra-mesh and Portal-oriented traffic scenarios. These increases cause the data traffic performance of the network created by PlainTC+ to approach the performance achieved when employing Max. Power. Thus, the performance gap to Max. Power is decreased despite achieving even greater cumulative transceiver power savings than PlainTC.

The PDR performance of PlainTC+ is contained in Tables 7.3 and 7.4 and the throughput results are contained in Tables 7.5 and 7.6.

#### 7.4.2.5 Resource Consumption

PlainTC+ was observed to consume 371KB of memory (2.3% of total memory) and the CPU utilisation equalled that of PlainTC at 0.3%. Therefore, the addition of the context-change mechanism did not significantly increase the recorded resource consumption.

TABLE 7.3: Average PDR Achieved for Intra-mesh Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC | PlainTC+ |
|---|---|---|---|
| 8 | 97.13 | 96.75 | 96.89 |
| 9 | 97.02 | 96.29 | 96.67 |
| 10 | 96.71 | 95.65 | 96.04 |
| 11 | 96.67 | 95.22 | 95.82 |
| 12 | 96.31 | 94.76 | 95.75 |
| 13 | 95.88 | 94.16 | 95.38 |
| 14 | 95.4 | 93.25 | 94.67 |

TABLE 7.4: Average PDR Achieved for Portal-oriented Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC | PlainTC+ |
|---|---|---|---|
| 8 | 99.72 | 99.46 | 99.48 |
| 9 | 99.66 | 99.53 | 99.56 |
| 10 | 99.53 | 99.15 | 99.35 |
| 11 | 99.36 | 98.95 | 99.02 |
| 12 | 99.19 | 98.44 | 98.62 |
| 13 | 98.98 | 98.12 | 98.52 |
| 14 | 98.71 | 98.02 | 98.26 |

TABLE 7.5: Average Throughput Achieved for Intra-mesh Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC | PlainTC+ |
|---|---|---|---|
| 8 | 328.17 | 327.54 | 327.75 |
| 9 | 263.91 | 263.85 | 263.89 |
| 10 | 227.43 | 221.50 | 223.12 |
| 11 | 213.87 | 211.09 | 211.95 |
| 12 | 196.51 | 180.82 | 188.52 |
| 13 | 192.63 | 184.25 | 185.65 |
| 14 | 185.11 | 170.92 | 177.35 |

TABLE 7.6: Average Throughput Achieved for Portal-oriented Traffic (Indoor Test-bed)

| Network Size | Max. Power | PlainTC | PlainTC+ |
|---|---|---|---|
| 8 | 313.57 | 312.27 | 312.55 |
| 9 | 271.82 | 270.83 | 271.02 |
| 10 | 219.65 | 209.71 | 212.65 |
| 11 | 192.72 | 184.18 | 188.63 |
| 12 | 164.57 | 159.73 | 161.88 |
| 13 | 155.79 | 151.52 | 151.92 |
| 14 | 147.35 | 131.61 | 137.55 |

## 7.5 Chapter Summary

This chapter has proposed a solution for reducing the topology instability caused by the PlainTC prototype. The instability is caused by the prototype's use of the CNN connectivity method and the resulting transceiver power changes made to maintain the required CNN. Changes to a node's transceiver power output level was shown to have a multiplicative effect on the number of changes observed for the neighbourhood size, network size and link quality variables when a global view of the network is considered. These four variables collectively represent a change to the context of a node and signify the topology instability caused by PlainTC.

A process for quantifying the contribution of each variable to the overall topology instability was provided in this chapter. This process employed a statistical method called Principal Component Analysis (PCA) in order to analyse historical network data collected every second over a 24-hour period using the indoor test-bed. This time period was sufficient to cover periods of network activity and inactivity and the data recorded every change to each context variable at each network node for the data-collection period. The PCA method substantiated the earlier assertion that a change to a node's transceiver power output level was a significant contributor to topology instability but also highlighted the contributions of the other context variables as well.

The quantified contributions of each variable to topology instability were subjected to a context-change threshold in order to reduce the number of changes made to a node's transceiver power output level. The motivation for the threshold value was to reduce the number of transceiver power changes by forcing nodes to react only to local events. The threshold value was therefore determined by considering both the frequency of changes reported for each variable as well as the contributions of each variable to the topology instability.

The PlainTC+ prototype was developed to incorporate the context-change mechanism and was subjected to a test-bed-based evaluation. The evaluation showed the anticipated reduction in the number of transceiver power changes and the subsequent reductions in the number of changes observed for the other context variables. The improvement in topology stability was shown to have a positive impact on network performance. The performance gap to the Max. Power scheme is narrowed and PlainTC+ offers a deployed I-WMN the possibility of achieving significant cumulative transceiver power savings with a minimal sacrifice in network performance.

The next chapter explores additional applications of the PlainTC and PlainTC+ prototypes within the domains of WMNs, WSNs and MANETs.

# Part III

# Further Applications and Conclusions

# Chapter 8

# Application of the Prototypes to WSNs and MANETs

## 8.1 Introduction

Part II of this thesis studied the dynamic real-time behaviours of PlainTC and PlainTC+ as well as their effects on network performance and energy efficiency. PlainTC was shown to cause topology instability which led to the development of PlainTC+. PlainTC+ was shown to reduce the topology instability and this reduction had a positive effect on the performance of the network as well as in the dynamic, real-time behaviour of the scheme.

This chapter explores possible applications of PlainTC and PlainTC+ outside of their originally-intended purpose. Further applications of these schemes in WMNs, WSNs, and MANETs can be found in Sections 8.2, 8.3 and 8.4 respectively.

## 8.2 Wireless Mesh Networks

Community-based WMN deployments are usually expected to grow in size as the benefits of using the network become common knowledge and new nodes are subsequently added to the network. WMNs are accommodating of the addition of new nodes and such additions are advantageous as they improve the network capacity.

The use of PlainTC or PlainTC+ in the I-WMN can help to monitor the expansion of the deployed network. The transceiver power assignment pattern produced by PlainTC[1] indicates that nodes close to the network centre employ lower transceiver power output levels than edge nodes. Edge nodes are often found to still employ the maximum power level in order to maintain the required CNN. Network expansion will typically occur at the network edge so as to increase the network's coverage area and add new participants. The effect of this expansion is that the current edge nodes will no longer remain so. Thus, PlainTC and PlainTC+ would cause a reduction in a former edge node's transceiver power output level as more neighbouring nodes are added to the backbone network. Network expansion can therefore be inferred in cases where a former edge node operates with a reduced transceiver power output level.

The advent of the use of the Software-Defined Networking (SDN) paradigm for future WMN deployments is a promising future avenue of use for PlainTC and PlainTC+. The SDN paradigm introduces a "smart" central controller that assumes the decision-making responsibility for the network. In an SDN-built I-WMN, the backbone nodes will act as the "dumb" implementation agents for the decisions taken at the controller. The controller will also possess the ability to tailor the decisions taken for individual backbone nodes and "push" the appropriate settings for each node.

Both the PlainTC and PlainTC+ schemes can be adapted to operate within the centralised SDN controller. All backbone nodes will be required to provide periodic status updates to the controller. These updates can be modified to include the input information required by both TC schemes. The centralised versions of each scheme can then use a global network view to determine whether a particular node should increase, decrease or maintain its current transceiver power output level.

[1] It is expected that PlainTC+ will exhibit a similar pattern as the context-change threshold does not regulate decreases in transceiver power output levels.

The PlainTC+ scheme will experience an additional benefit in an SDN-built WMN. The periodic reporting of all backbone nodes to the central controller will allow for dynamic adjustment of the context-change threshold value in response to prevailing network conditions. The context-change threshold value requires the quantification of the contributions of the four context variables to the observed topology instability. These contributions can be regularly calculated by the SDN controller by using the status updates being received from all the backbone nodes. The appropriate context-change threshold value for the prevailing network conditions can be subsequently determined in close-to-real-time and employed by the controller to manage the transceiver power output levels of the nodes in the I-WMN.

## 8.3 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are a form of ad-hoc network typically used for environmental monitoring. Nodes may contain several sensors, a processor and a wireless transceiver. These nodes are typically battery-powered as they are used to collect data in remote areas for prolonged periods of time. WSN nodes periodically transmit their collected data to a sink or gateway node for external collection and analysis. The large network size associated with WSNs requires multi-hop routes to transmit data from the sensor node to the sink node.

The battery-powered nature of WSNs has led to the use of sleep-based techniques and communication backbones to prolong node lifetimes. Nodes within close proximity to each other arrange themselves into clusters. This arrangement allows for some nodes to enter a sleep-state whilst neighbouring nodes remain awake and gathering data. Nodes periodically enter and exit their sleep-states in an effort to reduce their power consumption.

The cluster arrangement also results in the election of a clusterhead node that aggregates the data collected by the cluster's currently active nodes. The aggregated data is forwarded by the clusterhead to the sink node for bulk collection. Data forwarded by the clusterhead is routed along a communication backbone formed by other clusterhead nodes. Thus, a two-tier hierarchy is formed where data-gathering nodes represent the lower tier and the clusterheads represent the upper tier. The communication backbone formed by the clusterheads usually resembles a $k$-connected Connected Dominating Set (CDS).

TC has been previously considered for WSNs but this has primarily been in the form of regulating the sleep-states of the data-gathering nodes. PlainTC[2], in contrast, can be applied to the upper tier clusterhead nodes that form part of the communication backbone. These clusterhead nodes typically employ their maximum transceiver power output levels but this may not always be required.

The transceiver power output levels of the clusterhead nodes can be regulated by the number of neighbouring clusterhead nodes to be maintained. Thus, the battery depletion of the clusterhead nodes can be slowed by the transceiver power savings achieved the maximum transceiver power output level. This proposal is compatible with the existing sleep-based TC schemes used in WSNs. The sleep-based scheme can be used when nodes are fulfilling a data-gathering role but if the node is elected as a clusterhead then PlainTC can regulate the transceiver power output level used to connect to neighbouring clusterheads within the CDS backbone.

## 8.4 Mobile Ad-hoc Networks

Mobile ad-hoc networks (MANETs) are characterised by node mobility. Mobility either comes in the form of individual mobility or group mobility. PlainTC is better-suited to group mobility scenarios such as in tactical MANETs and vehicular ad-hoc networks (VANETs) as the node positions relative to each other do not change as often as in the individual mobility scenarios. Nodes in the group mobility scenarios typically employ the maximum transceiver power output levels and can benefit from employing PlainTC. PlainTC can use the CNN connectivity method to regulate the transceiver power outputs

[2]PlainTC+ is not recommended for WSNs as the method for quantifying context change and choosing an appropriate context-change threshold value may require too much effort to perform for nodes that are considered disposable due to their limited specifications and remote deployment. In addition, WSNs do not typically emphasise data traffic performance and its optimisation as network coverage, connectivity and energy-efficiency are greater concerns.

of the network nodes and potentially avoid the use of the maximum transceiver power level.

PlainTC can also benefit the tactical MANET and VANET scenarios as the reduced transceiver power output levels employed can reduce the probability of security threats such as eavesdropping. Eavesdropping occurs when an external actor collects transmitted packets for information gathering. A reduced transceiver power level means that an eavesdropper must be physically closer to the target node and therefore face the added risk of detection. Thus, although the information transmitted over these networks is likely to be encrypted, eavesdropping can allow a malicious actor to collect encrypted transmissions for subsequent decryption. Reductions in transceiver power output produced by PlainTC can thus make it harder to directly eavesdrop on target nodes as their communication ranges are reduced.

PlainTC also has the potential for course-grained position detection in the tactical MANET and VANET scenarios. Section 6.3.3 showed a correlation between the position of a node and its resultant transceiver power output level when using a CNN connectivity strategy. Nodes closer to the network centre were found to possess lower transceiver power output levels than the edge nodes. Despite the mobility present in group-based mobility scenarios, the nodes within a group do not move greatly relative to each other. Thus, the transceiver power assignment pattern is expected to be similar to that experienced within the I-WMN. The result is that the transceiver power level assigned to a node by PlainTC will be a good indicator of the node's position within its group of moving nodes.

## 8.5 Chapter Summary

This chapter has described additional applications of both the PlainTC and PlainTC+ schemes. The PlainTC scheme can be applied to both WSNs and MANETS whilst PlainTC+ can be used to monitor the expansion of currently-deployed community WMNs.

The next chapter summarises the key findings of the thesis and suggests future avenues of research.

# Chapter 9

# Conclusions and Future Work

TC has been reported to possess several benefits for wireless ad hoc networks, including WMNs. However, the actual efficacy of TC in WMNs has been difficult to determine due to the lack of data obtained from the study of TC prototypes. The lack of TC prototypes has been regarded in Santi (2005) as *"probably the most important open issue in this research field"*. According to Akyildiz and Wang (2009), the reason for this state of affairs is that *"many topology control schemes are proposed without considering the computation complexity or the possibility of being implemented. Although the performance is justified through theoretical analysis, they are difficult to be implemented"*. Numerous prototypes for Wireless Sensor Networks have since been developed but prototypes for WMNs have not been as forthcoming.

This thesis has focused upon the study of TC prototypes for the backbone networks of Infrastructure WMNs. Initial investigations of the two fundamental building blocks of a TC prototype for WMNs were presented in Chapter 3. The successful investigation resulted in the development of a TC prototype. The design and implementation of the PlainTC prototype are detailed in Chapter 4. The PlainTC prototype was subjected to both simulation- and test-bed-based evaluations and the details of the evaluation setups and measurement methodology were provided in Chapter 5. The viability of the PlainTC prototype as a mechanism for improving the energy-efficiency and network performance was assessed in Chapter 6. This assessment explored both the effect of PlainTC on the I-WMN as well as the dynamic, run-time behaviour of the prototype.

Chapter 6 also exposed a key shortcoming in the prototype's design. This shortcoming resulted in topology instability and would not have been discovered without a test-bed-based evaluation. The topology instability problem was addressed in Chapter 7, where contextual information was quantified and subjected to a threshold value in order to reduce the instability. An improved prototype, PlainTC+, was subjected to a further evaluation that demonstrated behavioural and performance improvements. Lastly, Chapter 8 provided additional uses of both PlainTC and PlainTC+ in WMNs, WSNs and MANETs.

## 9.1 Contributions

The goal of this thesis was to use a prototype to study the effectiveness of TC as a technique for improving the energy-efficiency and network performance of rural, battery-powered I-WMNs. The findings can however be applied to mains-powered I-WMN deployments. The case for employing the prototyping approach was motivated in Chapter 2 where it was shown that there is a lack of studies documenting the behaviour and performance of an autonomous TC prototype designed for the I-WMN. The main contributions claimed in this thesis are:

1. The establishment (via simulations and test-bed evaluations) that the Critical Number of Neighbours (CNN) connectivity strategy is a viable building block of a TC prototype for an I-WMN

2. An analysis of the design and implementation decisions taken to produce an IEEE 802.11s compatible, autonomous TC scheme suitable for low-cost, resource-constrained I-WMN nodes

3. An evaluation of the dynamic real-time behaviour of a TC prototype and its effectiveness as an energy efficiency and performance improvement mechanism

4. Context information can be employed to reduce the topology instability caused by a CNN-based TC prototype

**Contribution 1:** This Thesis enumerated the challenges to be faced when attempting to prototype a TC scheme for the I-WMN. These challenges allowed for alternate connectivity strategies to be eliminated until only the CNN strategy remained. A simulation- and test-bed-based evaluation of both fixed and adaptive CNN strategies was undertaken to assess the actual feasibility of using a CNN connectivity strategy as a building block of the TC prototype.

Candidate CNN strategies were subjected to connectivity, transceiver power output savings and convergence experiments. The evaluation results indicated that the CNN connectivity strategy defined in Xue and Kumar (2004) was a viable option for maintaining network connectivity whilst producing cumulative transceiver power savings. To the best of our knowledge, the test-bed-based evaluation of the various CNN connectivity strategies was the first such evaluation to be conducted.

**Contribution 2:** The exploratory experiments contained in Chapter 3 established that a CNN connectivity strategy can be used to maintain network connectivity whilst producing cumulative transceiver power savings. Chapter 3 also established that the popular Linksys WRT54GL wireless router platform was capable of transceiver power adjustment. These two findings provided the foundational building blocks for the design of the PlainTC prototype.

The design, found in Chapter 4, also considered the architectural constraints imposed by the node firmware, resource constraints imposed by the node hardware, the communication overhead caused by a distributed scheme and compatibility with the IEEE 802.11s standard that regulates the operation of the I-WMN.

The resultant prototype implementation produced a standards-compatible, autonomous scheme with three distinct advantages over existing WMN-applicable prototype implementation. These advantages were no additional communications overheads, avoiding the use of broadcast messages at multiple transceiver power output levels and avoiding the use of non-compliant routing tables that store the transceiver power level associated with each link to a neighbouring node. To the best of our knowledge, PlainTC is the only prototype that is both compatible with the IEEE 802.11s standard and does not require a re-engineering of existing data structures such as the routing table.

**Contribution 3:** PlainTC was subjected to both a testbed- and simulation-based evaluation. These evaluations assessed the viability of employing the CNN-based PlainTC scheme as a mechanism for improving the energy-efficiency and performance of an I-WMN.

The test-bed-based evaluation offered an opportunity to study the dynamic, real-time behaviour of PlainTC in response to a growing network. To the best of our knowledge, this represents the first instance of the study of a TC scheme's run-time behaviour. This knowledge gap in the literature allowed the study to establish benchmarks that can be used for future comparative studies of the dynamic behaviours of prototyped TC schemes.

PlainTC's resource consumption, and the delays in establishing both network connectivity and adjusting a node's transceiver power output to the appropriate level were observed. In addition, PlainTC's effect on the OLSR routing protocol and the effect of transceiver power adjustments were recorded.

Changes to a node's transceiver power output level were found to promote topology instability. Topology instability cannot be measured directly and was thus inferred from the number of observed changes to four context variables. These variables collectively represent the context of a node as they record changes to the node's transceiver power output level and its observed neighbourhood size, network size and link quality. Changes in at least one of the four observed variables result in changes to the underlying network topology. Changes in the network topology result in instability, which is detrimental to network performance.

The observation and analysis of PlainTC's effect on the stability of the network topology represents, to the best of our knowledge, the first instance of such an observation for a TC scheme. This observation was made possible by the use of a test-bed prototype and would have been prohibitively difficult to make with a simulation tool due to the tool's high levels of abstraction.

The simulation-based experiments showed that PlainTC degraded the energy-efficiency of the network despite achieving significant cumulative transceiver power savings. A correlation between the position of a node and its resultant transceiver power output level was also established. This represents a hitherto unique look at where the cumulative transceiver power savings originate.

The evaluation also showed that the network performance was degraded as a result of the cumulative transceiver power savings achieved. The power savings cause longer average path lengths that negatively affected the achieved PDR and throughput.

**Contribution 4:** The topology instability observed as part of the third contribution required a solution since a stable network topology was listed as a design criterion of the PlainTC prototype in Chapter 4. The analysis of the topology instability caused by PlainTC showed that the root cause was the adjustment of a node's transceiver power output levels.

Changes to a node's transceiver power output level were shown to have a multiplicative effect on the number of changes observed for the neighbourhood size, network size and link quality variables when a global view of the network is considered. These four variables collectively represent a change to the context of a node and signify the topology instability caused by PlainTC.

A process for quantifying the contribution of each variable to the overall topology instability was provided in Chapter 7. This process employed a statistical method called Principal Component Analysis (PCA) in order to analyse historical network data collected every second over a 24-hour period using the indoor test-bed. This time period was sufficient to cover both periods of network activity and inactivity and the data recorded every change to each context variable at each network node for the data-collection period. The PCA method substantiated the earlier assertion that a change to a node's transceiver power output level was a significant contributor to topology instability but also highlighted the contributions of the other context variables as well.

The quantified contributions of each variable to topology instability were subjected to a context-change threshold in order to reduce the number of changes made to a node's transceiver power output level. The motivation for the threshold value was to reduce the number of transceiver power changes by forcing nodes to react only to local events. The threshold value was therefore determined by considering both the frequency of changes reported for each variable as well as the contributions of each variable to the topology instability.

The PlainTC+ prototype was developed to incorporate the context-change mechanism and was subjected to a test-bed-based evaluation. The evaluation showed the anticipated reduction in the number of transceiver power changes and the subsequent reductions in the number of changes observed for the other context variables. Therefore a reduction in topology instability was achieved and gains in cumulative transceiver power savings, PDR and throughput were observed.

It must be remembered that the quantified values presented in Chapter 7 are specific to the indoor test-bed from which the analysed historical data was collected. However, the proposed process for quantifying context change can be conducted on any I-WMN deployment where changes to the four required context variables can be observed. The result will be quantified context change values that are tailored to the peculiarities of each network deployment.

## 9.2 Future Work

The work presented in this thesis provide potential avenues for further research. The next logical step is to evaluate both PlainTC and PlainTC+ on a community-based WMN deployment. A possible candidate deployment would be the outdoor, battery-powered deployment in Mankosi in the Eastern Cape province of South Africa. This network is described in Rey-Moreno et al. (2013). The benefit of such a deployment is that it provides for realistic traffic flows and the realistic energy consumption of the I-WMN nodes can be ascertained. Observing the effect that both proposed schemes have on the network performance and energy consumption of the deployment can lead to further insights being gained. These insights can lead to further improvements for TC in the I-WMN. The study of the proposed schemes in a pre-existing deployment does place additional importance on the design of such a study as the installation and subsequent data collection must cause minimal disruptions. This poses an additional research opportunity to establish best practice in the areas of network management and monitoring for ad hoc and wireless mesh networks.

Another promising avenue of research is the evaluation of the prototype schemes in group-mobility-based MANETs such as tactical MANETs and VANETs. These networks pose the ultimate test of the responsiveness of a TC scheme. The responsiveness of both PlainTC and PlainTC+ would be tested in two (2) ways: the speed with which an event that needs responding to is detected, and the speed at which these schemes execute their logic to determine the appropriate response. Both these aspects of responsiveness need to be minimised for the tactical MANET and VANET scenarios. Thus, opportunities for algorithm optimisation and source code optimisation exist. These optimisation opportunities may also extend to the WSN scenario where nodes with minimal computing resources are employed. The tactical MANET and VANET scenarios also provide an opportunity to consider the trade-off between the energy expended to respond to network events via transceiver power adjustment and the energy conserved if the power adjustment process results in a reduction of the transceiver power.

The third avenue of further research is the adaptation of the proposed context-change threshold mechanism for use in instances where TC is employed in other forms of ad hoc networking such as WSNs and MANETs. The suitability of the context variables that are currently employed (for an I-WMN) must be established for the WSN and MANET scenarios. In the event that these current context variables do not sufficiently account for the variability experienced by nodes in WSNs and MANETs then further work is required to identify those variables that do. Given the larger number of nodes associated with WSN and MANETs when compared to WMNs, this work may consider adopting a "big data" approach by collecting all the data made available by the node's firmware and the additional QoS mechanisms that may be employed. This data, collected over a period of time, may be mined to identify the collection of context variables most appropriate for use in the threshold mechanism for that particular type of ad hoc network. The proposed statistical method for determining the variable weightings can then be employed after the variable identification process has been completed. This data-driven approach to identifying context variables and their associated weightings may also lead to the development of predictive models for the dynamic, proactive adjustment of the variable weightings in anticipation of the occurrence of highly probable network events.

## 9.3 Closure

The increasing deployment of WMNs in developing regions requires greater research into applicable energy-efficiency and network performance mechanisms. TC was seen in the literature as an effective mechanism for improving both the energy-efficiency and performance of ad-hoc and mesh networks. The results obtained in this study suggest that this earlier reputation is undeserved as both prototypes underperformed against the current practice of nodes employing the maximum transceiver power level.

This work has provided a standards-compatible, autonomous TC prototype for the I-WMN and it is hoped that this prototype will serve as inspiration for future, improved prototypes. It is envisaged that future prototypes will achieve the twin goals of improving both the energy-efficiency and network performance of I-WMN deployments.

# Bibliography

Akyildiz, I., Wang, X., 2005. A survey on wireless mesh networks. IEEE Communications Magazine 43 (9), S23–S30.

Akyildiz, I., Wang, X., January 2009. Wireless Mesh Networks. Advanced Texts in Communications and Networking. Wiley.

Allen, W., Martin, A., Rangarajan, A., November 2005. Designing and deploying a rural ad-hoc community mesh network testbed. In: Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary. pp. –740–743.

Anastasi, G., Conti, M., Gregori, E., Passarella, A., 2008. 802.11 power-saving mode for mobile computing in wi-fi hotspots: Limitations, enhancements and open issues. Wireless Networks 14 (6), 745–768.

Ang, C., Tham, C., 2008. imst: A bandwidth guaranteed topology control algorithm for tdma-based ad hoc networks with sectorized antennas. Computer Networks 52 (9), 1675–1692.

Aron, F., Kurien, A., Hamam, Y., 2008a. A topology control algorithm for effective power efficiency and throughput for wireless mesh networks. In: Proceedings of the Third International Conference on Broadband Communications, Information Technology and Biomedical Applications. pp. 89–96.

Aron, F., Olwal, T., Kurien, A., Odhiambo, M., 2008b. A distributed topology control algorithm to conserve energy in heterogeneous wireless mesh networks. In: Proceedings of World Academy of Science, Engineering and Technology. pp. 530–536.

Aron, F., Olwal, T., Kurien, A., Odhiambo, M., August 2008c. Energy efficient topology control algorithm for wireless mesh networks. In: Proceedings of the International Wireless Communications and Mobile Computing Conference. pp. 135–140.

Aziz, A., Starobinski, D., Thiran, P., 2011. Understanding and tackling the root causes of instability in wireless mesh networks. IEEE/ACM Transactions on Networking 19 (4), 1178–1193.

Bahramgiri, M., Hajiaghayi, M., Mirrokni, V., October 2002. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In: Proceedings of the 11th International Conference on Computer Communications and Networks. pp. 392–397.

Balister, P., Bollobás, B., Sarkar, A., Walters, M., 2005. Connectivity of random k-nearest-neighbour graphs. Advances in Applied Probability 37 (1), 1–24.

Bash, L., Jelger, C., Tschudin, C., May 2007. A case study in designing an autonomic wireless mesh network. In: Proceedings of the 7th Scandinavian workshop on Wireless Ad-hoc Networks.

Blough, D., Leoncini, M., Resta, G., Santi, P., 2003a. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. Tech. Rep. IIT-TR-14/2003, Instituto di Informatica e Telematica, Pisa.

Blough, D., Leoncini, M., Resta, G., Santi, P., 2003b. The k-neighbors approach to physical degree bounded and symmetric topology control in ad hoc networks. Tech. Rep. IIT-TR-142003, Instituto di Informatica e Telematica, Pisa.

Borbash, S., Jennings, E., May 2002. Distributed topology control algorithm for multihop wireless networks. In: Proceedings of the 2002 International Joint Conference on Neural Networks. pp. 355–360.

Boushaba, M., Hafid, A., Gendreau, M., 2013. Local node stability-based routing for wireless mesh networks. In: Proceedings of the IEEE Wireless Communications and Networking Conference. pp. 1950–1955.

Bruno, R., Conti, M., Gregori, E., 2005. Mesh networks: commodity multihop ad hoc networks. IEEE Communications Magazine 43 (3), 123–131.

Burkhart, M., von Rickenbach, P., Wattenhofer, R., Zollinger, A., May 2004. Does topology control reduce interference? In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing. pp. 9–19.

Camp, J., Robinson, J., Steger, C., Knightly, E., June 2006. Measurement driven deployment of a two-tier urban mesh access network. In: Proceedings of the 4th International Conference on Mobile systems, applications and services. pp. 96–109.

Cardei, M., Wu, J., Yang, S., 2004. Topology control in ad hoc wireless networks with hitch-hiking. In: Proceedings of IEEE SECON. pp. 480–488.

Cheng, X., Narahari, B., Simha, R., Cheng, M., Liu, D., 2003. Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics. IEEE Transactions on Mobile Computing 2 (3), 248256.

Coates, R., Delwiche, M., 2009. Wireless mesh network for irrigation control and sensing. Transactions of the American Society of Agricultural and Biological Engineers 52 (3), 971–981.

Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, P., Schmidt, N., Zambonelli, F., 2006. A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems 1 (2), 223–259.

Donglan, Z., 2015. Research on secure network coding in wireless sensor networks based on web of things service. Open Automation and Control Systems Journal 7, 816–820.

Faheem, M., Abbas, M., Tuna, G., Gungor, V., 2015. Edhrp: Energy efficient event driven hybrid routing protocol for densely deployed wireless sensor networks. Journal of Network and Computer Applications.

Fainelli, F., November 2008. The openwrt embedded development framework. Tech. rep., http://downloads.openwrt.org/people/florian/fosdem/openwrt_cfp_fosdem2008.pdf.

Farbod, A., Todd, T., August 2007. Resource allocation and outage control for solar-powered wlan mesh networks. IEEE Transactions on Mobile Computing 6 (8), 960–970.

Freifunk, November 2008. Community – freifunk.net. World Wide Web, http://freifunk.net/community.

Gao, Y., Hou, J., Nguyen, H., 2008. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. In: Proceedings of the IEEE COnference on Computer Communications.

Hajek, B., March 1983. Adaptive transmission strategies and routing in mobile radio networks. In: Proceedings of the Seventeenth Annual Conference on Information Sciences and Systems. pp. 373–378.

Hall, P., 1985. On continuum percolation. The Annals of Probability 13, 1250–1266.

Hou, T., Li, V., 1986. Transmission range control in multihop packet radio networks. IEEE Transactions on Communications 34 (1), 38–44.

Hu, P., Hong, P., Li, J., Qin, Z., 2006. Tap: Traffic-aware topology control in on-demand ad hoc networks. Computer Communications 29 (18), 3877–3855.

Huang, Z., Shen, C., Srisathapornphat, C., Jaikaeo, C., 2002. Topology control for ad hoc networks with directional antennas. In: Proceedings of the International Conference on Computer Communications and Networks. pp. 16–21.

IEEE, 2008. Draft standard for information technology - telecommunications and information exchange between systems - lan/man specific requirements - part 11: Wireless medium access (mac) and physical layer (phy) specifications: Amendment: Mesh networking. Tech. Rep. P802.11s/D1.0, Institute of Electrical and Electronics Engineers.

Ishmael, J., Bury, S., Pezaros, D., Race, N., 2008. Deploying rural community wireless mesh networks. IEEE Internet Computing 12 (4), 22–29.

Jang, H., Fang, R., September 2008. Interference-aware topology control in wireless mesh network. In: Proceedings of the International Conference on Mobile Technology, Applications & Systems. pp. 1–7.

Jennings, E., Okino, C., 2002. Topology control for efficient information dissemination in ad-hoc networks. In: Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

Jia, X., Li, D., Du, D., March 2004. Qos topology control in ad hoc wireless networks. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communication Societies. pp. 1264–1272.

Johnson, D., 2007. Evaluation of a single radio rural mesh network in south africa. In: Proceedings of ICTD.

Johnson, D., 2013. Re-architecting internet access and wireless networks for rural developing regions. Ph.D. thesis, University of California Santa Barbara.

Jolliffe, I., 2002. Principal Component Analysis, 2nd Edition. Springer Series in Statistics. Springer-Verlag New York.

Kadivar, M., Shiri, M., Dehghan, M., 2009. Distributed topology control algorithm based on one- and two-hop neighbors information for ad hoc networks. Computer Communications 32 (2), 368–375.

Karyotis, V., Manolakos, A., Papavassiliou, S., 2010. On topology control and non-uniform node deployment in ad hoc networks. In: Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 522–527.

Kawadia, V., 2004. Protocols and architecture for wireless ad hoc networks. Ph.D. thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

Kawadia, V., Kumar, P., January 2005. Principles and protocols for power control in wireless ad hoc networks. IEEE Journal on Selected Areas in Communications 23 (1), 76–88.

Kleinrock, L., Silvester, J., December 1978. Optimum transmission radii for packet radio networks or why six is a magic number. In: National Telecommunications Conference. pp. 4.3.1–4.3.5.

Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M., 2000. The click modular router. ACM Transactions on Computer Systems 3 (18), 263–297.

Kowalik, K., Bykowski, M., Keegan, B., Davis, M., July 2008a. An evaluation of a conservative transmit power control mechanism on an indoor 802.11 wireless mesh testbed. In: Proceedings of the International Conference on Wireless Information Networks and Systems.

Kowalik, K., Bykowski, M., Keegan, B., Davis, M., June 2008b. Practical issues of power control in ieee 802.11 wireless devices. In: Proceedings of International Conference on Telecommunications. pp. 1–5.

Krebs, M., Stein, A., Lora, M., 2010. Topology stability-based clustering for wireless mesh networks. In: Proceedings of the Global Telecommunications Conference. pp. 1–5.

Kubo, H., Shinkuma, R., Takahashi, T., 2010. Topology control using multi-dimensional context parameters for mobile p2p networks. In: Proceedings of the IEEE Vehicular Technology Conference. pp. 1–5.

Labrador, M., Wightman, P., 2009. Topology Control in Wireless Sensor Networks: With a Companion Simulation Tool for Teaching and Research. Springer.

Le, S., Josse, J., Husson, F., 2008. Factominer: An r package for multivariate analysis. Journal for Statistical Software 25 (1), 1–18.

Li, L., Halpern, J., Bahl, P., Wang, Y., Wattenhofer, R., 2005a. A cone-based distributed topology-control algorithm for wireless multi-hop networks. IEEE/ACM TRANSACTIONS ON NETWORKING 13 (1), 147–159.

Li, N., Hou, J., April 2006. Localized fault-tolerant topology control in wireless ad hoc networks. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 17 (4), 307–320.

Li, N., Hou, J., Sha, L., April 2003. Design and analysis of an mst-based topology control algorithm. In: Proceedings of IEEE INFOCOM. p. 17021712.

Li, N., Hou, J., Sha, L., 2005b. Design and analysis of an mst-based topology control algorithm. IEEE Transactions on Wireless Communications 4 (3), 1195–1206.

Li, X., Wang, Y., Song, W., December 2004. Applications of k-local mst for topology control and broadcasting in wireless ad hoc networks. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 15 (12).

Liu, H., Zhang, B., Zheng, J., Mouftah, H., 2008. An energy-efficient localized topology control algorithm for wireless ad hoc and sensor networks. International Journal of Communication Systems 21, 1205–1220.

Lundgren, H., Ramachandran, K., Belding-Royer, E., Almeroth, K., Benny, M., Hewatt, A., Touma, A., Jardosh, A., 2006. Experiences from the design, deployment and usage of the ucsb meshnet testbed. IEEE Wireless Communications 13 (2), 18–29.

Mannweiler, C., Chakraborty, P., Klein, A., Schotten, H., 2012. Solarmesh - deployment aspects for wireless mesh networks in developing countries. In: Proceedings of 4th International ICST Conference, AFRICOMM. pp. 105–114.

Matos, R., Sargento, S., Hummel, K., Hess, A., Tutschku, K., Meer, H., 2012. Context-based wireless mesh networks: a case for network virtualization. Telecommunication Systems 51 (4), 259–272.

Mir, Z., Shrestha, D., Cho, G., Ko, Y., 2006. Mobility-aware distributed topology control for mobile multi-hop wireless networks. In: Chong, I., Kawahara, K. (Eds.), Proceedings of ICOIN 2006. Springer-Verlag, pp. 257–266.

Namboodiri, V., Gao, L., Janaswamy, R., 2008. Power-efficient topology control for static wireless networks with switched beam directional antennas. Ad Hoc Networks 6, 287–306.

Narayanaswamy, S., Kawadia, V., Sreenivas, R., Kumar, P., February 2002. Power control in ad hoc networks: theory, architecture, algorithm and implementation of the compow protocol. In: Proceedings of the European Wireless Conference 2002. Next

Generation Wireless Networks: Technologies, Protocols, Services and Applications. pp. 156–162.

Oki, O., 2013. Evaluating the effect of quality of service mechanisms in power-constrained wireless mesh networks. Master's thesis, The Department of Computer Science, Faculty of Science and Agriculture, University of Zululand.

openwrt.org, November 2008. Openwrt. World Wide Web, http://openwrt.org/.

Park, S., Sivakumar, R., November 2002. Load-sensitive transmission power control in wireless ad-hoc networks. In: Proceedings of the IEEE Global Telecommunications Conference. pp. 42–46.

Philips, T., Panwar, S., Tantawi, A., 1989. Connectivity properties of a packet radio network model. IEEE Transactions on Information Theory 35 (5), 770–777.

Ramachandran, K., Sheriff, I., Belding, E., Almeroth, K., 2007. Passive and Active Network Measurement: 8th Internatinoal Conference, PAM 2007, Louvain-la-neuve, Belgium, April 5-6, 2007. Proceedings. Springer Berlin Heidelberg, Ch. Routing Stability in Static Wireless Mesh Networks.

Raman, B., 2004. Digital gangetic plains: 802.11-based low-cost networking for rural areas, 2001-2004: A report. Tech. rep., DGP Media Labs Asia Team.

Ramanathan, R., Rosales-Hain, R., March 2000. Topology control of multihop wireless networks using transmit power adjustment. In: Proceedings of the IEEE INFOCOM. p. 404413.

Rey-Moreno, C., Roro, Z., Tucker, W., Siya, M., 2013. Community-based solar power revenue alternative to improve sustainability of a rural wireless mesh network. In: Proceedings of the Sixth International Conference on Information and Communications Technologies and Development. pp. 132–135.

Rodoplu, V., Meng, T., 1999. Minimum energy mobile wireless networks. IEEE Journal on Selected Areas in Communications 17 (8), 13331344.

Roslin, S., March 2015. Genetic algorithm based cluster head optimization using topology control for hazardous environment using wsn. In: Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). pp. 1–7.

Santi, P., 2005. Topology Control in Wireless Ad Hoc and Sensor Networks. Wiley.

Santi, P., Blough, D., 2003. The critical transmitting range for connectivity in sparse wireless ad hoc networks. IEEE Transactions on Mobile Computing 2 (1), 25–39.

Sen, S., Raman, B., 2007. Long distance wireless mesh network planning: problem formulation and solution. In: Proceedings of the 16th international conference on World Wide Web. pp. 893–902.

Sethu, H., Gerety, T., 2009. A new distributed topology control algorithm for wireless environments with non-uniform path loss and multipath propagation. Ad Hoc Networks.

Siripongwutikorn, P., Thipakorn, B., 2008. Mobility-aware topology control in mobile ad hoc networks. Computer Communications 31 (14), 3521–3532.

Song, W., Li, X., Frieder, O., Wang, W., 2006. Localized topology control for unicast and broadcast in wireless ad hoc networks. IEEE Transactions on Parallel and Distributed Systems 17 (4), 321–334.

Song, W., Wang, Y., Li, Y., Frieder, O., 2004. Localized algorithms for energy efficient topology in wireless ad hoc networks. In: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing. pp. 98–108.

Souihli, O., Frikha, M., Hamouda, M., 2009. Load balancing in manet shortest path routing protocols. Ad Hoc Networks 7 (2), 431–442.

Srivastava, G., Boustead, P., Chicharo, J., November 2004. Topology control in heterogeneous ad-hoc networks. In: Proceedings of the 12th IEEE International Conference on Networks. pp. 665–670.

Stallings, W., 2005. Wireless Communications and Networks, 2nd Edition. Prentice Hall.

Takagi, H., Kleinrock, L., 1984. Optimal transmission ranges for randomly distributed packet radio terminals. IEEE Transactions on Communications 32 (2), 246–257.

Tang, J., Xue, G., Zhang, W., 2005. Interference-aware topol-ogy control and qos routing in multi-channel wireless mesh networks. In: Proceedings of ACM MobiHoc. pp. 68–77.

tools for high-throughput data analysis, S. S., November 2015. Principal component analysis: How to reveal the most important variables in your data? - r software and data mining. World Wide Web, http://www.sthda.com/english/wiki/principal-component-analysis-how-to-reveal-the-most-important-variables-in-your-data-r-software-and-data-mining.

Valera, P., Lee, P., Wong, Y., Seah, W., Tan, H., Ju, H., November 2008. An experimental study on connectivity and topology control in real multi-hop networks. In: Proceedings of the fourth International Wireless Internet Conference.

Wan, P., Yi, C., May 2004. Asymptotic critical transmission radius and critical neighbor number for k-connectivity in wireless ad hoc networks. In: Proceedings of the fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing. pp. 1–8.

Wang, S., Wei, D., Kuo, S., 2006. An spt-based topology control algorithm for wireless ad hoc networks. Computer Communications 29, 3092–3103.

Wang, X., Lim, A., 2008. Ieee 802.11s wireless mesh networks: Frameworks and challenges. Ad Hoc Networks 6 (6), 970–984.

Wei, H., Rykowski, J., Dixit, S., 2013. WIFI, WIMAX, AND LTE MULTI-HOP MESH NETWORKS: Basic Communication Protocols and Application Areas. WILEY SERIES ON INFORMATION AND COMMUNICATION TECHNOLOGY. Wiley.

Wu, D., Gupta, D., Mohapatra, P., 2011. Qurinet: A wide-area wireless mesh testbed for research and experimental evaluations. Ad Hoc Networks 9 (7), 1221–1237.

Wu, K., Liao, W., January 2006. Interference-efficient topology control in wireless ad hoc networks. In: IEEE Consumer Communications and Networking Conference. pp. 411–415.

Xue, F., Kumar, P., 2004. The number of neighbors needed for connectivity of wireless networks. Wireless Networks 10 (2), 169–181.

Zhang, J., Yan, H., Wang, B., Wang, J., Fu, X., 2015. Self-adaptive sleep scheduling for wireless sensor networks. International Journal of Wireless and Mobile Computing 8 (4), 346–352.

# Part IV

# Appendices

# PlainTC code for the Linksys WRT54GL router

```
#!/bin/sh
#PlainTC


# FUNCTIONS

log(){                          #
   x=$1;                   #
   n=2;
   l=-1;                   #  log n1 n2 = log n2 to the base of n1
   if [ $2 != "" ]; then           #
      n=$x                     #
      x=$2                     #
      echo "x is" $x
   fi                          #
   #while [ $(($x)) -ge 1 ]; do
   while [ "$x" -ge 1 ]; do
      #let
         $((l=$(($l))+1))
         $((x=$(($x))/$(($n))))
         #x/=$n;
         echo "test"
   done
   return_val=$l
   if [ $return_val -eq 0 ]; then
      echo "inside log"
      return_val=$return_val+1
   fi
   echo "the result of the log is "$(($(($return_val))))

   #echo $l
   return
}

adjustTxPower() {
   echo "Adjusting tx power level"
   echo nvram set wl0_txpwr=\"$(($current_tx_power_level))\"
   #$(nvram set wl0_txpwr="$((current_tx_power_level))")

   $(nvram set wl0_txpwr="$(($current_tx_power_level))")

   #$(nvram set wl0_txpwr=\"$current_tx_power_level\")

   $(nvram commit)
   $(wifi)
   $(wl txpwrlimit)
   return
}

# VARIABLES
##################################################################
####################################################
```

```
x=0
l=0
return_val=0

no_network_nodes=$(($(($(route | grep "metric" -c))+1)) # determining the number of
nodes available at the Network Layer.
echo "The number of network nodes is "$no_network_nodes

# Upper bounds for the various connectivity strategies
xue_kumar=51774                    #
wan_yi=27180                       #
balister_et_al_1=5139              # all values multiplied by 10000 because of restriction to
integer arithmetic only
balister_et_al_2=9967              #
ferrari_tonguz_1=31400             #
worst_case=$(($no_network_nodes-1))
santi_blough=9 # the number of neighbours to be maintained is 9
#ferrari_tonguz_2=$((1+0)) # this value requires a base of 2 in the log calculation

###########################################################################
#########################################################

defaultCNN=6
tx_power_change_level=3 # the amount in qdBm that we either increase or decrease the tx_power
max_tx_power_level=78 #qdBm / 89mW
min_tx_power_level=1  #qdBm
initial_tx_power_level=$(nvram get wl0_txpwr) #stores the initial tx power level and used to
compare with the current tx power level
echo "The initial tx power is" $initial_tx_power_level
current_tx_power_level=$(nvram get wl0_txpwr) #stores the current tx_power level
n=$(wget -q -O - http://localhost:2006/neighbours|sed -e's/Table: Links//;s/Local
IP//;s/remote IP//;s/Hysteresis//;s/LinkQuality//;s/lost//;s/total//;s/NLQ//;s/ETX//' |
grep "10.1.*" -c) #temporary value to store the number of occurrences of 10.1.*.*
current_no_neighbours=$(($n/2)) #get the current number of neighbours
echo "The current number of neighbours is" $current_no_neighbours


# CODE
#if [ $current_tx_power_level -eq $max_tx_power_level ]; then
   #exit
#fi

if [ $no_network_nodes -eq 0 ]; then # checking if node is isolated at Network Layer
   current_tx_power_level=$max_tx_power_level # set tx power to max to help establish
Network Layer connectivity
   adjustTxPower
   #exit
fi
if [ $no_network_nodes -gt 0 ]; then # checking if node is connected at Network Layer

   if [ $(($current_no_neighbours)) -gt $(($no_network_nodes)) ]; then
      no_network_nodes=$(($current_no_neighbours))
   fi
```

```
    log 10 $no_network_nodes # function determining the log of the total number of nodes visible at
the Network Layer
    CNN_x=$(($(($xue_kumar*$(($(($return_val))))))+10000)) #used to circumvent
inability to multiply real numbers eg. x.yz. We add 10000 to compensate for a lack of precision.
    CNN=$(($(($CNN_x))/10000)) # The division results in an integer result which is reasonably
close to the expected integer result.

    #CNN=$(($worst_case))

    echo "The CNN is " $(($(($CNN))))
    if [ $(($current_tx_power_level)) -le $(($max_tx_power_level)) ]; then
        echo "inside if 1"
        if [ $(($current_no_neighbours)) -gt $(($CNN)) ]; then #tx power needs to be
adjusted downwards
            echo "inside if 2"
            if [ $(($(($(current_tx_power_level))-$((tx_power_change_level)))) -ge $((
$min_tx_power_level)) ]; then # determining whether an adjustment is feasible
                echo "inside if 3"
                current_tx_power_level=$(($current_tx_power_level-
$tx_power_change_level)) #
                echo "The current tx power level is" $current_tx_power_level "qdBm"
        #setting a new tx power
                adjustTxPower                                           #
            fi
        fi
        if [ $(($current_no_neighbours)) -lt $(($CNN)) ]; then #tx power needs to be
adjusted upwards
            echo "inside if 4"
            if [ $(($current_tx_power_level+$tx_power_change_level)) -le $((
$max_tx_power_level)) ]; then # determining whether an adjustment is feasible
                echo "inside if 5"
                current_tx_power_level=$(($current_tx_power_level+
$tx_power_change_level)) #
                echo "The current tx power level is" $current_tx_power_level "qdBm"
        # setting a new tx power
                adjustTxPower                                           #
            fi
        fi
        if [ $current_no_neighbours -eq $(($CNN)) ]; then
            #do nothing
            echo "do nothing"
        fi
    fi
fi
```

# PlainTC code for the Atarraya simulator

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package atarraya.event;

import atarraya.constants;
import atarraya.element.candidate;
import atarraya.element.node;
import atarraya.element.register;
import atarraya.atarraya_frame;

import java.lang.Math;
import java.util.*;

/**
 *
 * @author Pragasen
 */

public class EventHandlerTCPlainTC implements EventHandlerSW,constants{

    int type= TC_PROTOCOL;

    atarraya_frame father;

    int active;
    int inactive;
    int initial;
    int sleeping;
    boolean TM_Selected;
    boolean SD_Selected;
    boolean COMM_Selected;
    int TMType;
    int temp_selectedTMprotocol;
    int temp_num_structures;
    int k_value;
    int numNodes;
    double avgNeighb;                    // Average number of neighbors of a node on the topology
    int contNeighb;                   // Temporary variable to calculate the average number of neighbors

    /** Creates a new instance of EventHandlerCDS */
    public EventHandlerTCPlainTC(atarraya_frame _frame)
    {
        //gets the network size
        System.out.println("The number of network nodes is: " + atarraya_frame.
numnodes);
        ////////////////////////////////////////////////////////////////////
        //code to determine the node degree before PlainTC
        int sum = 0;
        for(int j=0;j<atarraya_frame.numnodes;j++){
            sum = sum + getNode(j).getNeighbourList().size();
```

```
        }
        if(sum != 0){
            System.out.println("The average node degree at max power is: " + (sum/
atarraya_frame.numnodes));
        }
        ////////////////////////////////////////////////////////////////////

        father = _frame;

        TM_Selected=false;
        SD_Selected=false;
        COMM_Selected=false;
        TMType = NO_TM;

        if(father.getVariable(SELECTED_TM_PROTOCOL)!=TM_PROTOCOL_NO_SELECTED)
            TM_Selected=true;

        if(father.getVariable(SELECTED_SD_PROTOCOL)!=
SENSOR_PROTOCOL_NO_SELECTED)
            SD_Selected=true;

        if(father.getVariable(SELECTED_COMM_PROTOCOL)!=
COMM_PROTOCOL_NO_SELECTED)
            COMM_Selected=true;


        //k_value = (int)father.getVariable(KNEIGH_K);

        //determines the CNN to be maintained:
        k_value = (int)(5.1774 * (Math.log10((double)(atarraya_frame.numnodes))));
        //k_value = 9;
        //k_value = (int)(2.718 * (Math.log10((double)(atarraya_frame.numnodes))));
        System.out.println("The CNN is: " + k_value);

        temp_selectedTMprotocol = (int)father.getVariable(SELECTED_TM_PROTOCOL);

        temp_num_structures = (int)father.getVariable(NUMINFRASTRUCTURES);
        System.out.println("temp_num_structures = "+ temp_num_structures);

        active = 0;
        inactive = 0;
        initial = 0;
        sleeping = 0;
    }

    public int getTMType(){
        return TMType;
    }

    public void setLabels(int _initial,int _active, int _inactive, int _sleeping){
        active=_active;
        //inactive=_inactive;
        //initial=_initial;
```

```
        //sleeping = _sleeping;
    }

    public int getTreeID()
    {
        return father.getTreeID();
    }

    public int getSortingMode(){
        return (int)father.getVariable(SORTINGMODE);
    }

    public int getSimMode()
    {
        return (int)father.getVariable(SIMMODE);
    }


     public int getbatchMode(){
        return (int)father.getVariable(BATCH_SIMULATION);
    }

     public node getNode(int id)
    {
        return father.getNode(id);
    }

    public void pushEvent(event_sim e)
    {
        father.pushEvent(e);
    }

    public void AddTreeLine(int s, int d, int t)
    {
        //father.AddTreeLine(s,d,t);
    }

    public void RemoveTreeLine(int s, int d, int t)
    {
        //father.RemoveTreeLine(s,d,t);
    }

    public void frame_repaint()
    {
        father.frame_repaint();
    }

    public void InvalidateAllEventsFromIDFromTimeTOfCodeC(int id, double t, int c, int
tree){
        father.InvalidateAllEventsFromIDFromTimeTOfCodeC(id,t,c,tree);
    }

    public void InvalidateAllEventsFromIDFromTimeT(int id, double t, int tree){
```

```
        father.InvalidateAllEventsFromIDFromTimeT(id,t,tree);
    }

    public void InvalidateAllEventsFromIDFromTimeTOfTypeTy(int id, double t, int Ty,
int tree){
        father.InvalidateAllEventsFromIDFromTimeTOfTypeTy(id,t,Ty,tree);
    }

    public void InvalidateAllEventsFromIDFromTimeTOfTypeTyOfCodeC(int id, double t,
int Ty, int c, int tree){
        father.InvalidateAllEventsFromIDFromTimeTOfTypeTyOfCodeC(id,t,Ty,c,tree);
    }

 public void broadcast(double final_time, double current_time, int _sender, int
_destination, int _code){
        father.broadcast(final_time, current_time, _sender, _destination, _code,"",-1,type);
    }

 public void broadcast(double final_time, double current_time, int _sender, int
_destination, int _code, int _tree){
        father.broadcast(final_time, current_time, _sender, _destination, _code,"",_tree,
type);
    }

 public void broadcast(double final_time, double current_time, int _sender, int
_destination, int _code, String msg){
        father.broadcast(final_time, current_time, _sender, _destination, _code,msg,type);
    }

 public void broadcast(double final_time, double current_time, int _sender, int
_destination, int _code, String msg, int _tree){
        father.broadcast(final_time, current_time,-1,-1, _sender, _destination, _code,msg,
_tree,type);
    }

 public void broadcast(double final_time, double current_time, int _source, int
_final_destination,int _sender, int _destination, int _code, String msg, int _tree){
        father.broadcast(final_time, current_time,_source,_final_destination, _sender,
_destination, _code,msg,_tree,type);
    }

    public double getRandom(double max_val)
    {
        return java.lang.Math.random()*max_val;
    }

    public void showMessage(String m)
    {
        father.showMessage(m);
    }
```

```
    public void init_nodes(int _treeID){
        int i;
        double _clock = father.getVariable(CLOCK);

        //System.out.println("PlainTC: NUMPOINTS: " + father.getVariable(NUMPOINTS)); //: Pragasen
        for(i=0;i<father.getVariable(NUMPOINTS);i++){

            getNode(i).setState(initial,_treeID);
            getNode(i).defineLabels(initial,active, inactive,sleeping);
            getNode(i).SetInfrastructureStarted(_treeID,true);
            pushEvent(new event_sim(_clock+(10*TIMEOUT_DELAY_SHORT), _clock, i, i,
TIMEOUT_RECOGNITION,"",_treeID,type));

             if(TM_Selected){
                //Program the next topology creation
                pushEvent(new event_sim(_clock+getRandom(PROCESSING_DELAY), _clock,
i, i, RESET_TM_PROTOCOL, "",_treeID,TM_PROTOCOL));
            }

            if(SD_Selected){
                //stop all future event from the sensor-data protocol
                pushEvent(new event_sim(_clock+getRandom(PROCESSING_DELAY), _clock,
i, i, RESET_QUERY_SENSOR, "",_treeID,SENSOR_PROTOCOL));
            }

        }
    }

    public void init_node(int _treeID, int id_node){
        getNode(id_node).setState(S_NOT_VISITED,_treeID);

        performPlainTC(id_node);
    }

    public void performPlainTC(int id_node){ //all PlainTC's logic can be found here
        if(id_node >= atarraya_frame.numnodes){
            //getNode(id_node).cleanNeighbors();
            System.out.println("This is a SINK node: " + id_node);
        }
        else{
            System.out.println("Performing PlainTC on a REGULAR node: " + id_node);
            ////////////////////////////////////////////////////////////////////////////////
            Vector neighbourList, newNeighbourList;// = null;
            //for(int count=0; count < atarraya_frame.numpoints; count++){
            node currentNode = getNode(id_node);
            int neighbourListSize = currentNode.getNeighbourList().size();
            System.out.println("REGULAR node " + id_node + " has " + neighbourListSize +
" neighbours.");
            int neighbourListDifference=0;

            neighbourList = currentNode.getNeighbourList();
            for(int j = 0; j < neighbourListSize; j++){
                try{
```

```
            if(currentNode.getNeighborID(j) >= atarraya_frame.numnodes){
                try{
                    //System.out.println("test");
                    neighbourList.removeElementAt(j); //removes sink nodes from neighbour list.
Pragasen
                }
                catch(Exception e){
                    e.printStackTrace();
                }
            }
            }catch(Exception f){
                f.printStackTrace();
            }
        }

        neighbourListSize = currentNode.getNeighbourList().size();//determine updated no.
of neighbouring nodes
        if(neighbourListSize > k_value){
            newNeighbourList = currentNode.getNeighbourList();

            System.out.println("The current node is: " + id_node + " and it has " +
neighbourListSize +  " elements in its neighbour list");

            neighbourListDifference = neighbourListSize - k_value; //determines the number
of neighbouring nodes to remove (via a reduction of tx power/range)
            System.out.println("The neighbour list difference is: " +
neighbourListDifference);

            for(int j = 0; j < neighbourListDifference; j++){
                newNeighbourList.removeElementAt(0); //simulates the adjustment of tx
power/range by removing the first j neighbours (these neighbours are the furthest away)

                //System.out.println(currentNode.getNeighbourList().elementAt(j));
            }

            neighbourListSize = currentNode.getNeighbourList().size(); //determine the no.
of neighbours after applying PlainTC
            for(int j = 0; j < neighbourListSize; j++){
                System.out.println(currentNode.getNeighbourList().elementAt(j));
            }

            //@todo: still need to adjust node's new tx power: Pragasen

            //cannot use the BuildNeighborhood method in the atarraya_frame class since it builds the
neighbourhood on max power.
            //must modify the local BuildNeighbourhood method to accept the distance of furthest
neighbour and
            //use this value to build the new neighbourhood.

            node farthestNode = getNode(currentNode.getNeighborID(0));
            double x1 = currentNode.getPosition().getX();
            double y1 = currentNode.getPosition().getY();
            double x2 = farthestNode.getPosition().getX();
```

```
            double y2 = farthestNode.getPosition().getY();
            double dist = java.lang.Math.round(java.lang.Math.sqrt(Math.pow(x1-x2,2
)+Math.pow(y1-y2,2)));
            System.out.println("The farthest node is "+ farthestNode.getID() + ", " + dist
 + "m away.");
            currentNode.setRadius((int)dist);
            currentNode.setNodeRadius(dist);

            avgNeighb += buildNeighbourhood(currentNode.getID(), dist);
            //atarraya_frame.myPanel.repaint();
         }
         /////////////////////////////////////////////////////////////////
         //code to determine the average node degree after PlainTC
         int sum = 0;
         for(int i=0;i<atarraya_frame.numnodes;i++){
            sum = sum + getNode(i).getNeighbourList().size();
         }
         if(sum != 0){
            System.out.println("The average node degree after PlainTC is: " + (sum/
atarraya_frame.numnodes));
         }
      /////////////////////////////////////////////////////////////////
      }
      //father.myPanel.repaint();
      //}
   }

   public void initial_event(int _id, int _treeID){
      getNode(_id).setState(C_BLACK,_treeID);
      getNode(_id).setLevel(0, _treeID);
      double _clock = father.getVariable(CLOCK);
      pushEvent(new event_sim(_clock+PROCESSING_DELAY, _clock, _id, _id,
SEND_HELLO, "",_treeID,type));
      //father.simulation_pause();
      getNode(_id).addGateway(_treeID,_id,_id);
      getNode(_id).setSinkAddress(_treeID, _id);
      getNode(_id).setDefaultGateway(_treeID,_id);

      if(SD_Selected){
         //stop all future event from the sensor-data protocol
         pushEvent(new event_sim(_clock+getRandom(PROCESSING_DELAY), _clock,
_id, _id, RESET_QUERY_SENSOR, "",_treeID,SENSOR_PROTOCOL));
      }
   }


   public boolean CheckIfDesiredFinalState(int s){
      //if(s==C_BLACK_FINAL || s==C_UNMARKED)
      //if(s==active || s==inactive || s==sleeping)
      if(s==active)
         return true;

      return false;
```

```
    }

    public boolean CheckIfDesiredSleepingState(int s){
        //if(s==C_UNMARKED)
//      if(s==sleeping)
//          return true;

        return false;
    }

    public boolean verifIfNodeInCDS(int _id, int _treeID){

        //if(getNode(_id).getState(_treeID)==C_BLACK_FINAL || _treeID==-1)
        if(getNode(_id).getState(_treeID)==active || _treeID==-1)
            return true;

        return false;
    }

    public int GetMessageSize(int code){
        return SIZE_SHORT_PACKETS;
    }

    public void HandleEvent(event_sim e){

        int code = e.getCode();
        int sender = e.getSender();
        int source = e.getSource();
        int final_destination = e.getFinalDestination();
        int receiver = e.getReceiver();
        int destination = e.getDestination();
        double temp_clock = e.getTime();
        String temp_data = e.getData();
        int temp_tree = e.getTree();
        int temp_data_int=0;
        int temp_data_int2=0;
        int temp_data_int3=0;
        double temp_data_double=0.0;
        String[] temp_data_array = new String[50];
        String[] temp_data_array2;
        String data, data1 = "TRANSITION", data2 = "WHITE";


        int numneighb,i,x,temp_ID;
        int index,tam;
        int min_child;
        register temp_reg;
        double tempdist;
        int temp_level=0;
        int temptam=0;
        int tempmax,nmax;
        int temp=0;
        int temp2=0;
```

```
        int temp3=0;
        int temp4=0;
        int temp5=0;
        double temp_time;
        double metric=0;
        double weight;
        double temp_weight;
        boolean sw;
        boolean sw2;
        candidate temp_cand;
        double temp_data_double2;


        switch(code){

            case INIT_NODE:
                init_node(temp_tree, receiver);
                //performPlainTC(receiver);
                break;

            case INIT_EVENT:
                if(getNode(receiver).isSink())
                    initial_event(receiver,temp_tree);
                //performPlainTC(receiver);
                break;

            //The node receives a signal to send its first hello message
            case SEND_HELLO:
                broadcast(temp_clock+getRandom(MAX_TX_DELAY_RANDOM), temp_clock,
sender, -1, HELLO, ""+getNode(sender).getLevel(temp_tree)+"#"+getNode(sender).
getSinkAddress(temp_tree),temp_tree);
                pushEvent(new event_sim(temp_clock+TIMEOUT_DELAY, temp_clock, sender
, sender, TIMEOUT_RECOGNITION,"",temp_tree,type));
                break;

            //The node receives a hello message
            case HELLO:

                //performPlainTC(receiver);

                temp=0;

                 temp_data_array = temp_data.split("#");
                try{

                    temp = Integer.parseInt(temp_data_array[0]);        //temp = Level of the node
it received the Hello message

                    temp2 = Integer.parseInt(temp_data_array[1]);        //temp2 = sink's address'
                }
                catch(Exception ex)
                    {ex.printStackTrace();
                }
```

```
            if(getNode(receiver).getState(temp_tree) == initial){
                pushEvent(new event_sim(temp_clock+PROCESSING_DELAY, temp_clock,
receiver, receiver, SEND_HELLO, "",temp_tree,type));
                getNode(receiver).setState(KN_VISITED,temp_tree);

                getNode(receiver).setLevel(temp+1, temp_tree);
                //getNode(receiver).setLevel(temp, temp_tree);//: Pragasen

                System.out.println("new level is: "+ (temp));
                System.out.println("temp_tree: " + temp_tree);
                System.out.println("temp2: " + temp2);


                //getNode(receiver).setDefaultGateway(temp_tree, sender);
                getNode(receiver).setSinkAddress(temp_tree, temp2);
                InvalidateAllEventsFromIDFromTimeTOfTypeTyOfCodeC(receiver,
temp_clock, type, TIMEOUT_RECOGNITION, temp_tree);
            }
            break;

            //When the Hello timeout expires
        case TIMEOUT_RECOGNITION:
            System.out.println("Timeout Recognition");

            //performPlainTC(receiver);
            performPlainTC(sender);
            temptam = getNode(sender).getNumCandidates(temp_tree);
            //System.out.println("temptam: "+ temptam);
            temp2=Math.min(temptam,k_value);
            temp_data = "";
            sw=true;

            if(temp2>0){

                //Sort candidates based on the distance
                getNode(sender).SortCandidates(temp_tree);


                i=0;

                while(i<temp2){

                    //Get the i th candidate
                    temp3 = getNode(sender).getCandidate(temp_tree,i).getID();

                    i++;
                }


                //Get the distance to the k th node or the parent node
                temp_weight = getNode(sender).getCandidate(temp_tree, i-1).getMetric();
                System.out.println("Node " + getNode(sender).getID() + "'s k-th
```

```
neighbour is " + temp_weight + "m away." );

                //Assign the greatest distance
                getNode(sender).setRadius(temp_tree, Math.ceil(temp_weight));



                ////////////////////////////////////////////////////////////////////////////////
                //Neighbour List after PlainTC
                for(int k = 0; k < getNode(receiver).getNeighbourList().size(); k++){
                   System.out.println("The neighbour list for node " + getNode(k).getID()
+ " is: ");
                   //System.out.println(getNode(k).getID());
                   System.out.println(getNode(receiver).getNeighbourList().get(k));
                }
                System.out.println();
                ////////////////////////////////////////////////////////////////////////////////
             }

             getNode(sender).setState(KN_UPDATED,temp_tree);

             getNode(sender).setParentState(temp_tree, true);

             //Initiate the TM protocol only on the active nodes of the topology, except on the local
             if(TM_Selected && (father.getTMType() != TM_ENERGY || getNode(sender).
isSink(temp_tree))){
                pushEvent(new event_sim(temp_clock+DELTA_TIME, temp_clock, receiver,
 receiver, INIT_EVENT, "",temp_tree,TM_PROTOCOL));
             }

             //Initiate the sensor querying protocol only on the active nodes of the topology!!
             if(SD_Selected  && getNode(receiver).getActiveTree() == temp_tree){
                pushEvent(new event_sim(temp_clock+DELTA_TIME, temp_clock, receiver,
 receiver, INIT_EVENT, "",temp_tree,SENSOR_PROTOCOL));
             }

             //Initiate the COMM  protocol only on the active nodes of the topology!!
             if(COMM_Selected){
                pushEvent(new event_sim(temp_clock+DELTA_TIME, temp_clock, receiver,
 receiver, INIT_EVENT, "",temp_tree,COMM_PROTOCOL));
             }



             break;




       }
```

```
    }



    public int buildNeighbourhood(int nodeID, double farthest_neighbour_distance){
      int i;
      contNeighb=0;

     atarraya_frame.myPanel.removeAllLinesFromSource(nodeID);

      //Remove all previous neighbors from the node's data structure
      getNode(nodeID).cleanNeighbors();

      //Regenerate the neighborhood on the max power graph
      for(i=0;i<atarraya_frame.numnodes;i++){
          if(i!=nodeID){
          //if(getNode(nodeID).isNeighbor2(getNode(i))){
          if(getNode(nodeID).isNeighbour3(getNode(nodeID), getNode(i),
farthest_neighbour_distance)){
              if(!atarraya_frame.batch_creation)
                 atarraya_frame.myPanel.addLine(nodeID,i);
              contNeighb++;
              }
          }
          //}
      }

      getNode(nodeID).SortNeighbors();
      return (int)contNeighb;

   }
}
```

# PlainTC+ code for the Linksys WRT54GL router

```
#!/bin/sh -x
#PlainTC+

# FUNCTIONS

log(){                                  #
    x=$1;                          #
    n=2;
    l=-1;                          #  log n1 n2 = log n2 to the base of n1
    if [ $2 != "" ]; then                    #
        n=$x                         #
        x=$2                         #
        echo "x is" $x
    fi                             #
    #while [ $(($x)) -ge 1 ]; do
    while [ "$x" -ge 1 ]; do
        #let
            $((l=$(($l))+1))
            $((x=$(($x))/$(($n))))
            #x/=$n;
            echo "test"
    done
    return_val=$l
    if [ $return_val -eq 0 ]; then
        echo "inside log"
        return_val=$return_val+1
    fi
    echo "the result of the log is "$(($(($return_val))))

    #echo $l
    return
}

adjustTxPower(){
    echo "Adjusting tx power level"
    echo nvram set wl0_txpwr=\"$(($current_tx_power_level))\"
    #$(nvram set wl0_txpwr="$((current_tx_power_level))")

    $(nvram set wl0_txpwr="$(($current_tx_power_level))")

    #$(nvram set wl0_txpwr=\"$current_tx_power_level\")

    $(nvram commit)
    $(wifi)
    $(wl txpwrlimit)
    return
}

calculateContextChange(){
    echo "Calculating the context change threshold"
    #context_change_threshold=0.5 # this threshold controls when node transceiver power can be
increased. This value can be changed.
```

```
previous_noise=$(nvram get previous_noise)
if [ -z "$previous_noise" ]; then
   echo "inside if 6"
   previous_noise=0
fi
previous_rssi=$(nvram get previous_rssi)
if [ -z "$previous_rssi" ]; then
   previous_rssi=0
fi
previous_no_network_nodes=$(nvram get previous_no_network_nodes)
if [ -z "$previous_no_network_nodes" ]; then
   previous_no_network_nodes=0
fi
previous_neighbourhood_size=$(nvram get previous_neighbourhood_size)
if [ -z "$previous_neighbourhood_size" ]; then
   previous_neighbourhood_size=0
fi
previous_tx_power=$(nvram get previous_tx_power)
if [ -z "$previous_relative_position" ]; then
   previous_tx_power=0
fi

temp1=$(($(($current_no_network_nodes))-$(($previous_no_network_nodes))))
if [ $(($temp1)) -eq 0 ]; then
   network_size_change=0
fi
if [ $(($temp1)) -ne 0 ]; then
   network_size_change=1
fi
echo "network size change = "$network_size_change


temp1=$(($(($current_neighbourhood_size))-$(($previous_neighbourhood_size))))
if [ $(($temp1)) -eq 0 ]; then
   neighbourhood_size_change=0
fi
if [ $(($temp1)) -ne 0 ]; then
   neighbourhood_size_change=1
fi
echo "neighbourhood size change = "$neighbourhood_size_change


echo "current noise is: "$current_noise
echo "current rssi is: "$current_rssi
temp2=$(($(($current_noise))-$(($current_rssi))))
temp3=$(($(($previous_noise))-$(($previous_rssi))))
temp1=$(($(($temp2))-$(($temp3))))

if [ $(($temp1)) -eq 0 ]; then
   link_quality_change=0
fi
if [ $(($temp1)) -ne 0 ]; then
   link_quality_change=1
```

```
    fi
    echo "link quality change = "$link_quality_change


    temp1=$(($(($current_tx_power))-$(($previous_tx_power))))

    if [ $(($temp1)) -eq 0 ]; then
        tx_power_change=0
    fi
    if [ $(($temp1)) -ne 0 ]; then
        tx_power_change=1
    fi
    echo "relative tx power change = "$tx_power_change


    current_context_change=$((($(($2482*$(($network_size_change)))))+($((2767
*$(($neighbourhood_size_change)))))+($((1956*$(($link_quality_change)))))+($((
2795*$(($tx_power_change)))))))
    echo "context change value = "$current_context_change

    #previous_noise=$current_noise
    $(nvram set previous_noise="$(($current_noise))") # store previous noise value in nvram for
future comparisons
    previous_rssi=$current_rssi
    $(nvram set previous_rssi="$(($current_rssi))") # store previous rssi value in nvram for
future comparisons
    previous_no_network_nodes=$current_no_network_nodes
    $(nvram set previous_no_network_nodes="$(($current_no_network_nodes))") # store
previous network size in nvram for future comparisons
    previous_neighbourhood_size=$current_neighbourhood_size
    $(nvram set previous_neighbourhood_size="$(($current_neighbourhood_size))") #
store previous neighbourhood size in nvram for future comparisons
    previous_tx_power=$current_tx_power
    $(nvram set previous_tx_power="$(($current_tx_power_level))") # store previous
relative position in nvram for future comparisons

    $(nvram commit)
    return
}

# VARIABLES
##################################################################
####################################################
x=0
l=0
return_val=0
current_context_change=0

no_network_nodes=$(($(($(route | grep "metric" -c)))) # determining the number of nodes
available at the Network Layer.
echo "The number of network nodes is "$no_network_nodes

# Upper bounds for the various connectivity strategies
```

```
xue_kumar=51774                    #
wan_yi=27180                       #
balister_et_al_1=5139               # all values multiplied by 10000 because of restriction to
integer arithmetic only
balister_et_al_2=9967               #
ferrari_tonguz_1=31400              #
worst_case=$(($no_network_nodes-1))
santi_blough=9 # the number of neighbours to be maintained is 9
#ferrari_tonguz_2=$((1+0)) # this value requires a base of 2 in the log calculation
defaultCNN=6

# Transceiver Power Variables
tx_power_change_level=3 # the amount in qdBm that we either increase or decrease the tx_power
max_tx_power_level=78 #qdBm / 89mW
min_tx_power_level=1  #qdBm
initial_tx_power_level=$(nvram get wl0_txpwr) #stores the initial tx power level and used to
compare with the current tx power level
echo "The initial tx power is" $initial_tx_power_level
current_tx_power_level=$(nvram get wl0_txpwr) #stores the current tx_power level
n=$(wget -q -O - http://localhost:2006/neighbours|sed -e's/Table: Links//;s/Local
IP//;s/remote IP//;s/Hysteresis//;s/LinkQuality//;s/lost//;s/total//;s/NLQ//;s/ETX//' |
grep "10.1.*" -c) #temporary value to store the number of occurrences of 10.1.*.*

# Current number of neighbours
current_no_neighbours=$(($n/2)) #get the current number of neighbours
echo "The current number of neighbours is" $current_no_neighbours

# Context change variables
context_change_threshold=7205# this threshold controls when node transceiver power can be
increased. This value can be changed.Multiplied by 10000 to avoid float arithmetic
echo "The context change threshold is "$context_change_threshold

current_noise=$(wl noise|sed -e's/noise is //') # current noise does not need to be stored in
nvram because it can be retrived via wl
current_rssi=$(wl rssi|sed -e's/rssi is //') # current rssi does not need to be stored in nvram
because it can be retrived via wl
current_no_network_nodes=$no_network_nodes
current_neighbourhood_size=$current_no_neighbours
current_tx_power=$current_tx_power_level

#################################################################
###################################################

# CODE
#if [ $current_tx_power_level -eq $max_tx_power_level ]; then
    #exit
#fi

if [ $no_network_nodes -eq 0 ]; then # checking if node is isolated at Network Layer
    current_tx_power_level=$max_tx_power_level # set tx power to max to help establish
Network Layer connectivity
    adjustTxPower
    #exit
```

```
fi
if [ $no_network_nodes -gt 0 ]; then # checking if node is connected at Network Layer

   if [ $(($current_no_neighbours)) -gt $(($no_network_nodes)) ]; then
      no_network_nodes=$(($current_no_neighbours))
   fi


   log 10 $(($(($no_network_nodes))+1)) # function determining the log of the total number of
nodes visible at the Network Layer
   CNN_x=$(($(($xue_kumar*$(($(($return_val))))))+10000)) #used to circumvent
inability to multiply real numbers eg. x.yz. We add 10000 to compensate for a lack of precision.
   CNN=$(($(($CNN_x))/10000)) # The division results in an integer result which is reasonably
close to the expected integer result.

   #CNN=$(($worst_case))

   echo "The CNN is " $(($(($CNN))))
   if [ $(($current_tx_power_level)) -le $(($max_tx_power_level)) ]; then
      echo "inside if 1"
      if [ $(($current_no_neighbours)) -gt $(($CNN)) ]; then #tx power needs to be
adjusted downwards
         echo "inside if 2"
         if [ $(($((current_tx_power_level))-$((tx_power_change_level)))) -ge $((
$min_tx_power_level)) ]; then # determining whether an adjustment is feasible
            echo "inside if 3"
            current_tx_power_level=$(($current_tx_power_level-
$tx_power_change_level)) #
            #there is no need to determine the magnitude of context change when reducing a node's
transceiver power.
            adjustTxPower #
            #fi
            echo "The current tx power level is" $current_tx_power_level "qdBm"
      #setting a new tx power
         fi
      fi
      if [ $(($current_no_neighbours)) -lt $(($CNN)) ]; then #tx power needs to be
adjusted upwards
         echo "inside if 4"
         if [ $(($current_tx_power_level+$tx_power_change_level)) -le $((
$max_tx_power_level)) ]; then # determining whether an adjustment is feasible
            echo "inside if 5"
            calculateContextChange
            if [ $(($current_context_change)) -ge $(($context_change_threshold)) ];
then # Transceiver power is only increased if context change is greater than the pre-defined threshold
               current_tx_power_level=$(($current_tx_power_level+
$tx_power_change_level)) #
               adjustTxPower                                     #
               echo "The current tx power level is" $current_tx_power_level "qdBm"
      # setting a new tx power
            fi
         fi
      fi
```

```
        if [ $(($current_no_neighbours)) -eq $(($CNN)) ]; then
            #do nothing
            echo "do nothing"
        fi
    fi
fi
```

# R commands for calculating variable weights

Step 1

```
install.packages("FactoMineR")
devtools::install_github("kassambara/factoextra")
library("factoextra")
mydata = read.csv("context.csv")
```

Step 2

```
library("FactoMineR")
res.pca <- PCA(mydata, graph = FALSE)
print(res.pca)
```

Step 3

```
eigenvalues <- res.pca$eig
head(eigenvalues[, 1:3])
```

Step 4

```
head(res.pca$var$contrib)
```