Reliable transmission power control for Internet of Things

Citation for published version (APA):
Kotian, R. (2017). *Reliable transmission power control for Internet of Things*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven.

Document status and date:
Published: 19/06/2017

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

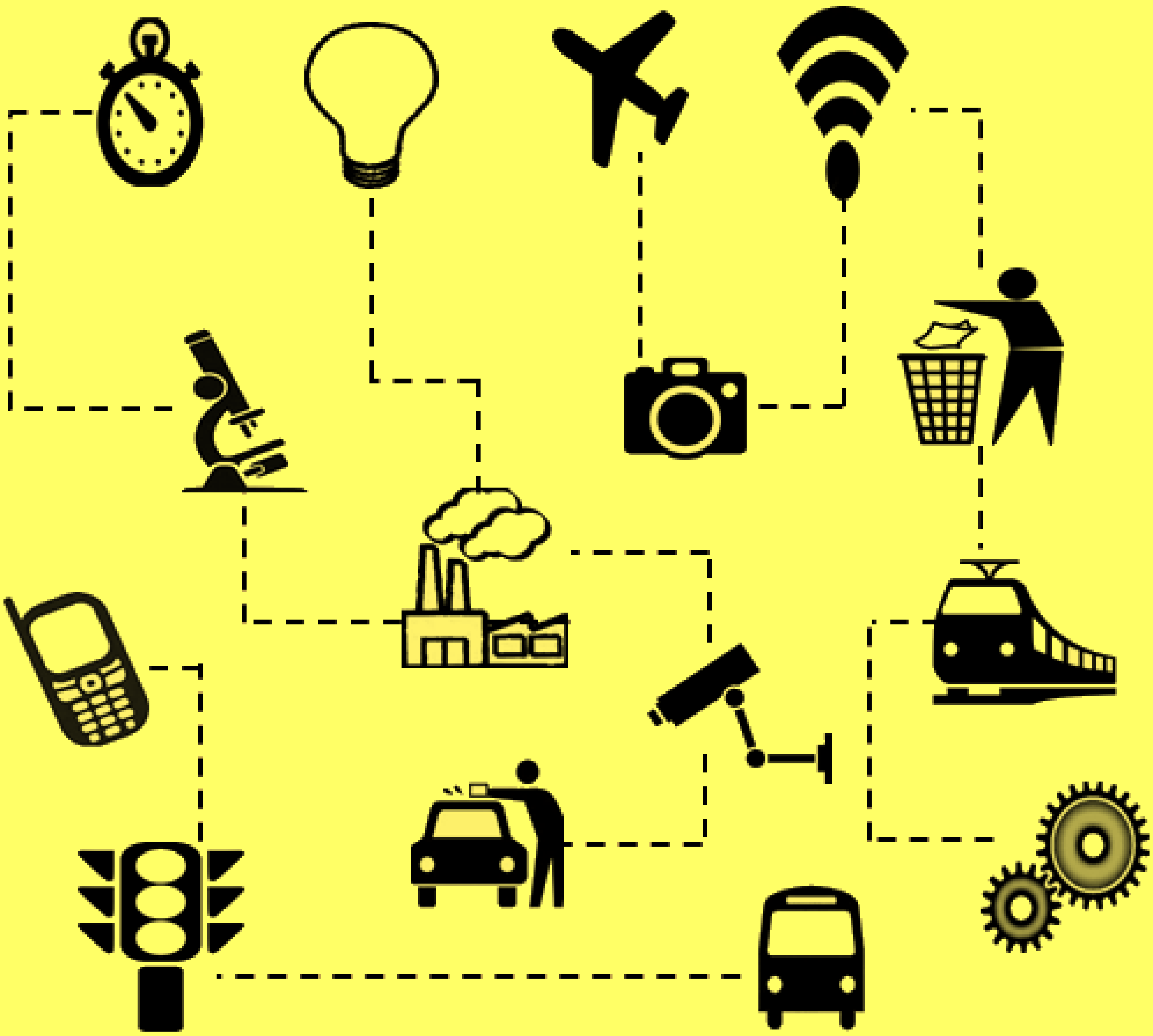
Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 08. Feb. 2024

Reliable Transmission Power Control for Internet of Things



Roshan Kotian

Reliable Transmission Power Control for Internet of Things

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. F.P.T Baaijens, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op maandag 19 juni 2017 om 16.00 uur

door

Roshan Kotian

geboren te Karnataka, India

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter: prof.dr.ir. A.B. Smolders
le promoter: prof.dr. A. Liotta
copromotor: dr.G. Exarchakos

leden: prof.dr. J.F. Martínez (Universidad Politécnica de Madrid)
prof.dr. L. Liu (University of Derby)
prof.ir. A.M.J. Koonen
prof.dr.ir. I.G.M.M. Niemegeers
prof.dr.ir. J.P.M.G. Linnartz

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

This thesis has been approved by a committee with the following members:

Voorzitter: prof.dr.ir. A.B. Smolders
le promoter: prof.dr. A. Liotta
copromotor: dr.G. Exarchakos

leden: prof.dr. J.F. Martínez (Universidad Politécnica de Madrid)
prof.dr. L. Liu (University of Derby)
prof.ir. A.M.J. Koonen
prof.dr.ir. I.G.M.M. Niemegeers
prof.dr.ir. J.P.M.G. Linnartz

A catalogue record is available from the Eindhoven University of Technology Library

Title: Reliable Transmission Power Control for Internet of Things
Author: Roshan Kotian
Eindhoven, Technische Universiteit Eindhoven, 2017
ISBN: 978-90-386-4314-4
NUR: 980

Keywords: Wireless Sensor Network / Transmission Power Control/ Internet of Things / Routing Protocols / MAC Protocols

Copyright © 2017 by Roshan Kotian
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means without the prior written consent of the author.

“Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things”

Isaac Newton

Summary

Reliable Transmission Power Control for Internet of Things

The Internet of Things (IoT) is revolutionizing many sectors such as logistics, transportation, manufacturing plants, and agriculture by improving their operational efficiency, energy usage, and management of resources. The IoT network as a whole is projected to scale at the tunes of trillion of nodes within the next few years. It is estimated that on an average 57% of the large-scale IoT network will be deployed to environmental monitoring, the transportation sector, smart city, and drones.

Major impediments to the successful operation of this complex interconnected system are limited energy availability of the individual nodes and ever-changing radio propagation medium. The uncertainty in the channel conditions is due to interference from other sources such as Wi-Fi routers, microwave ovens. In addition, natural obstacles such as office walls decay the signal strength. Furthermore, environmental conditions such as humidity and temperature in the indoor and outdoor locations significantly weaken the communication link reliability. Poor reliability also has other adverse effects such as reduction in the lifetime of the node, as packets may have to be retransmitted. To enhance the link reliability between a pair of sensor nodes, boosting transmission power level seems to be a natural choice. However, the transceiver of several sensor nodes such as TelosB, MicaZ etc. consumes more energy than other units such as ROM, sensors, and CPU of the nodes. In addition, it increases the contention in the network.

Faced with the difficulty of prolonging the network lifetime and at the same time increasing the network reliability, one of the solutions is to employ a Transmission Power Control (TPC) technique that scales the transmission power up or down at run-time whenever the link quality falls below or above a predefined threshold respectively. Therefore, the main objective of a TPC algorithm is to achieve optimal transmission power –a power level that does not break the already established link between a pair of nodes nor increase the contention in the network.

The decision to change transmission power level based on the values of inexpensive hardware metrics such as Received Signal Strength Indicator (RSSI) or Link Quality Indicator (LQI) is not appropriate. This is because they are sensitive to environmental disturbances and the probability of it deviating from a predefined threshold level remains high. This causes respective nodes in the network to change its transmission power level frequently. This fluctuation in the power level in a dense network consisting of multiple transmitting nodes increases the degree of interference resulting in a collision and ultimately negatively influences the reliability and results in higher energy usage as the packets are retransmitted when the collision occurs.

This thesis, therefore, focusses on five problems. First, it investigates the impact of one node performing TPC in the network. The simulation experiment reveals that when the duration of the fluctuation in transmission power level is longer and random, it increases the interference and hence the retransmission. The other noted side effects are increased in the latency and higher energy consumption. The thesis also discusses the reason for the degradation in the network metrics such as Packet Delivery Ratio (PDR), latency and energy consumption.

Second, this thesis provides a detailed study on the impact of fixed minimum and maximum transmission power on a sizeable static network and then discusses the concept of optimal transmission power level.

Third, the thesis proposes a proactive Data Aware Transmission Power Control (DA-TPC) algorithm that reduces the interference caused due to prolonged duration of changes in the transmission power level. Unlike other TPC algorithm such as Adaptive and Robust Topology Control (ART) that is reactive to the fall in the link quality and has a longer power fluctuation period, proactive TPC such as DA-TPC predicts the quality of the link in advance and quickly selects the optimum power level on per packet basis. The simulation experiment show that compared to reactive TPC, proactive TPC scheme increases the reliability of the network.

Fourth, the thesis proposes to employ the priority of the data as a metric to be used as an input to the TPC algorithm instead of other sensitive and erroneous metrics such RSSI and LQI. The application layer tags the data that it receives from the sensors with a specific priority. Based on the tagged information appropriate transmission power level is activated at the lower layers. For instance, the temperature of the room is classified as a high priority data once it reaches an unusual level. If the temperature of the room is within an expected predefined level it is classified as a low priority data. Comparative analysis of metric shows that the energy consumption is lower and the reliability between a pair of nodes and the entire network are significantly higher when TPC algorithm uses priority of the data as a metric instead of RSSI.

Generally, as transceiver consumes more energy, it is worthwhile to predict the energy level of sensor nodes so that the remaining energy level could be used as one of the decision points to adjust the transmission power level. Therefore, the thesis finally investigates the applicability of RSSI as an input to various state-of-the-art supervised machine-learning algorithms such as Support Vector Machine (SVM), Logistic Regression (LR) etc. for an early and accurate detection of energy depletion rate. The experiments conducted shows that the nature of RSSI values can aid the machine learning algorithms to predict the fall in the energy level only when they reach cut-off level (1.6V to 1.5V) –a minimum operational voltage required for the normal operation of the TelosB sensor motes. When the energy level of the nodes reaches the cut-off level, the speed and the prediction accuracy of SVM and LR is the highest, whereas Linear Regression and Random Classifier has the worst performance compared to all other algorithms evaluated.

Contents

Summary iv
Chapter 1 Internet-of-Things: Oppurtunities and Challenges 11
  1.1 Introduction 11
  1.2 Definition of IoT 12
  1.3 Harnessing the Power of IoT 12
  1.4 Overview of Technological Concerns in Realizing IoT 15
  1.5 Device Connectivity: A Bigger Challenge 16
    1.5.1 Internal and External Interference 16
    1.5.2 Environmental Factors 17
    1.5.3 Ad-hoc Deployment 17
    1.5.4 Software and Hardware Configuration 17
  1.6 Research Objectives 18
  1.7 Experimental Configuration 20
    1.7.1 IoT Device 20
    1.7.2 IoT Operating System 21
    1.7.3 Simulation Environment 22
    1.7.4 Networking Protocol 22
    1.7.5 Routing and MAC Protocols 23
  1.8 Contributions and Organization of the Thesis 24
Chapter 2 TPC Framework: Background and Insights 27
  2.1 General Framework of TPC 27
  2.2 Classification of Link Quality Metrics 28
    2.2.1 Hardware Based Metrics 29

2.2.2 Software Based Metrics....30
2.3 Classification of TPC Algorithms....31
2.3.1 Non-Machine Learning Approaches....31
2.3.2 Machine Learning Approaches....33
2.4 Limitations of TPC Algorithms....34
2.4.1 Drawbacks of Hardware based Link Quality Metrics....35
2.4.2 Drawbacks of Software based Link Quality Metrics....36
2.4.3 Drawbacks of TPC Algorithms....36
2.5 Conclusions....39
Chapter 3 Impact of TPC in Multi-Hop Networks....41
3.1 Related Work....42
3.1.1 General Limitations....42
3.1.2 Implications of TPC on MAC Layer....43
3.1.3 Implications of TPC on Routing Layer....43
3.1.4 Limitations of Popular TPC....44
3.2 Background....46
3.2.1 Generic Transmission Power Control Model....46
3.2.2 MAC Layer Protocol: ContikiMAC....47
3.2.3 Routing Layer Protocol: ContikiMesh and CTP....48
3.3 Experimentation and Simulation Setup....50
3.3.1 Constraints and Requirements....50
3.3.2 Network Topology....51
3.3.3 Transmission Power Scaling....52
3.3.4 Emulation Features....56
3.3.5 Benchmark and Assessment Metrics....57
3.4 Results and Discussions....58
3.4.1 Transmission Power Scaling over ContikiMesh....58
3.4.2 Transmission Power Scaling over CTP....64
3.4.3 Transmission Power impact on ContikiMesh and CTP....68

3.5 Conclusions and Future Work 69

Chapter 4 Reliable Low-Power Wireless Networks over Unstable Transmission Power 71

4.1 Link Quality at Low Transmission Power 72

4.2 Related Work 74

4.3 Drawbacks of Low Transmission Power 75

4.3.1 Latency is higher 75

4.3.2 Energy-Hole problem 75

4.3.3 Hidden-Node problem 75

4.3.4 Breakage of Low-Power Link is higher 76

4.4 Experimental Setup and Simulation Parameters 76

4.4.1 General Characteristics 77

4.4.2 Network Deployment 77

4.4.3 Protocol Stack Configuration 80

4.5 Results and Discussion 81

4.5.1 Linear Deployment of 15 nodes 81

4.5.2 Hybrid Deployment of 15 nodes 85

4.6 Conclusion and Future Work 89

Chapter 5 Assessment of Non-Machine Learning Algorithms for Transmission Power Control 91

5.1 Drawbacks of Employing Machine-Learning for TPC 92

5.2 Experimental Set-up 93

5.3 Characteristics of Input Data 95

5.4 Overview of Algorithms Evaluated 96

5.5 Performance Evaluation Metrics 98

5.6 Evaluation of Algorithms 99

5.6.1 Performance of SMA and WMA 99

5.6.2 Performance of EWMA 100

5.6.3 Performance of Linear Regression 101

5.6.4 Performance of Kalman Filter 101
5.7 Conclusions 102
Chapter 6 Data Aware Transmission Power Control (DA-TPC) 105
6.1 Drawbacks of TPC Model 106
6.2 Overview of DA-TPC 108
6.3 Components of DA-TPC 109
6.3.1 Working of Initialization Component of DA-TPC 110
6.3.2 Working of Anomaly Detection Component 112
6.3.3 Working of Routing Component 113
6.3.4 DA-TPC 113
6.4 Experimental and Simulation Setup 113
6.5 Results and Discussion 115
6.5.1 Performance of DA-TPC at Network Level 115
6.5.2 Performance of TPC at Node Level 117
6.5.3 Energy Consumption at Initialization Phase 119
6.6 Conclusions and Future Work 120
Chapter 7 Predicting the Energy Depletion of IoT Devices 122
7.1 Importance of Predicting Energy Level 123
7.2 Literature Overview 124
7.3 System Overview 126
7.4 Experimental Setup 128
7.5 Overview of Machine-Learning Algorithms 129
7.6 Evaluation 131
7.7 Conclusions and Future Work 136
Chapter 8 Conclusions 138
References 142
List of Publications 155
Curriculum Vitae 157

Chapter 1 Internet-of-Things: Oppurtunities and Challenges

1.1 Introduction

Internet-of-Things (IoT) is revolutionizing many sectors such as logistics and transportation, manufacturing plants and agriculture by increasing their operational efficiency, reducing the energy usage and better management of resources. Many forward-looking companies are embracing the IoT to attain higher performance or to generate values like never before. As IoT provides immense benefits, it's becoming a competitive advantage and the business houses have to consider it as a part of their strategic plan.

As a result, the IoT network as a whole is projected to scale at the tunes of billions of nodes within the next few years. As the number of connected devices increases exponentially, achieving higher network capacity and reliability with lower latency and energy consumption is challenging. It is estimated that the IoT will cause the Internet Protocol (IP) traffic to increase by 300% by 2018 [1].

Although not all the network embedded devices (e.g. sensor nodes) of IoT network will have to simultaneously communicate among one another, there will typically be hundreds of nodes within relevant multi-hop cluster networks. Thus, it is crucial to look at how reliability, the latency of IoT network is affected in high-density communication scenarios with existing communication protocols. Furthermore, these sensor nodes comprising IoT networks operate on a limited battery power. Remote deployment of an IoT network makes it difficult for the field technician to replace the battery sources. Hence, prolonging the battery life of the sensor nodes is necessary. Therefore, the main focus of this thesis is to address the following

How can we enhance the communication reliability of IoT network with Transmission Power Control (TPC)?

One of the major techniques we consider to address this question is to use Transmission Power Control (TPC) algorithms that aid every individual node of the network to employ appropriate transmission power level at run-time. A well designed TPC algorithm can not only reduce the energy consumption but also has the potential to improve the reliability of the network.

In the remainder of this chapter, we first provide the definition of IoT in section 1.2. A brief overview of the real world deployment of IoT network is provided in section 1.3. In Section 1.4, we highlight some of the major issues that may reduce the operational efficiency of a large-scale IoT networks. The section 1.5 provides a detailed explanation on how the reliability of the IoT network can be disrupted by various factors. In section 1.6, we break down the broader research question mentioned above into more detailed research problems and summarize each of the research problems investigated in this thesis. A brief explanation of experimental platform, software, and hardware utilized for the experimentation is described in section 1.7. Contributions and organization of the thesis are outlined in section 1.8.

1.2 Definition of IoT

The term Internet-of-Things (IoT) is not well defined and therefore has been loosely used as a buzzword in scientific research and marketing strategies. One can find the various definition of IoT. However, we adopt the following definition as it provides a better perspective on the work carried out in this thesis

The Internet of Things could be conceptually defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "thing" have identities, physical attributes, and virtual personalities, use intelligent interfaces, and are seamlessly integrated into the information network [2].

1.3 Harnessing the Power of IoT

Many organizations are seeing the value of adopting IoT for as diverse reasons as saving cost, improving service, and finding new revenue streams. As IoT provides unique opportunities, Cisco estimates that there will be around 50 billion IoT devices by 2020 [3]. The growth of IoT devices as shown Figure 1.1 from its inception in 2009 is phenomenal.

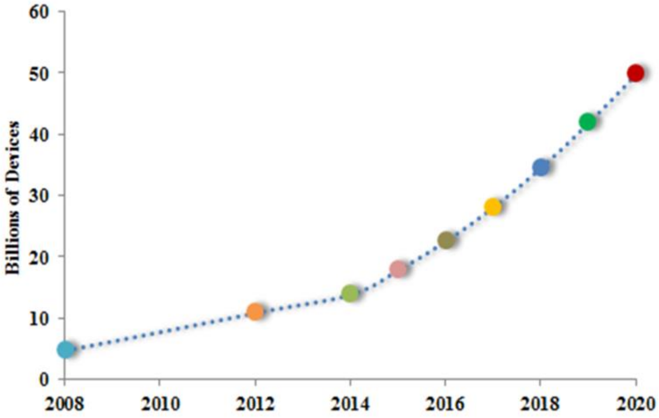


Figure 1.1 Growth of IoT [4]

Figure 1.2 is derived from a recent survey indicating that a majority of the survey respondents expressed their keen interest in designing, implementing and deploying IoT applications and network for home automation, environmental monitoring, and search & rescue operation through drones [5]. From Figure 1.2, we can interpret that there is an equal amount of interest amongst the various stakeholders to deploy IoT infrastructure for both indoor and outdoor scenarios.

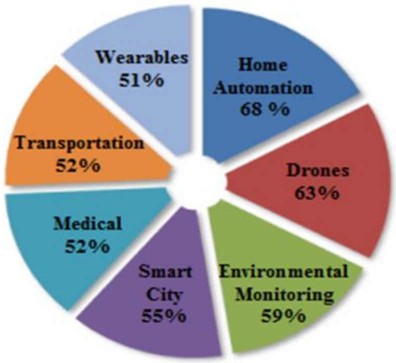


Figure 1.2 Interest of IoT in various domains

Figure 1.3 depicts the value-at-stake for various sectors [6]. According to Cisco, value-at-stake is the bottom-line value (higher revenues and lower costs) that can be created or will migrate among the companies and the industries based on their ability to harness IoT [6]. Figure 1.3 depicts five major business domains that can substantially increase their value-at-stake by employing IoT. Amongst the five major domains, we find that the impact of IoT on factories or manufacturing units is the highest. Incorporating IoT to manufacturing processes has the potential to minimize raw material and energy wastages. The addition of devices with the connectivity and inferencing capability leads to an intelligent device that offers a better control of the machinery equipment and can assist the humans to improve the product quality.

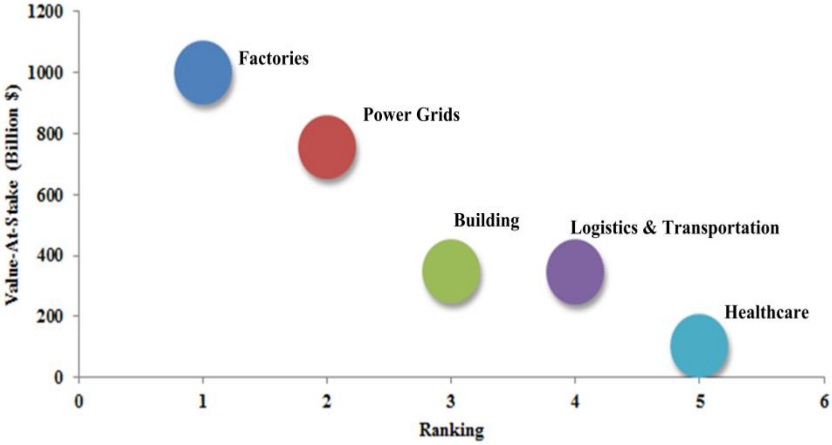


Figure 1.3 Potential value-at-stake for various sectors [4]

The second domain that stands to benefit from the IoT is a power grid. The current power grid installations are not efficient as they are incapable of generating power based on the real-time need. Embodying IoT that can perform sensing and control, can detect the supply and demand misalignment.

Architectural and Building business unit is the third in the list that can create higher value-at-stake. HVAC are the major source of energy consuming devices in public institutions such as office complexes, hospitals, hotels etc. According to General Electrics, making HVAC smart through IoT connectivity can save around 5% of the energy in a small-sized industrial power plant generating 15MW [7].

Logistics and Transportation group is the fourth business domain that can increase its value-at-stake. Traffic congestion, fuel consumption can greatly be reduced by vehicles that are connected to their surrounding (e.g. traffic signals, tollbooths) through IoT.

Lastly, medical and healthcare business also can enhance their value-at-stake. It is estimated that 4 to 17% of the hospitalized patients suffer from a life threatening events such as cardiac or respiratory arrests [8]. It is also found that 70% of such events could be prevented [8]. Therefore, a continuous remote monitoring is the need of the hour. IoT with its connectivity and real-time monitoring functionalities can solve this pressing problem. Added benefit of remote continuous monitoring through IoT enables the patient care to be performed in less expensive setting such as a home.

Clearly, IoT with its sensing, monitoring, inferencing and communication capabilities will empower the end users to interpret a vast amount of data and react to the changing environment. The economic potential of IoT is significant and moving forward, it is only going to grow.

1.4 Overview of Technological Concerns in Realizing IoT

Although IoT has the potential to save time and resources of many business houses, it has its own set of unique problems. Figure 1.4 provides the information about the major concerns of the respondents in a recent survey [5]. As Figure 1.4 shows, security, and privacy are two of the biggest concerns. IoT brings with it a wide array of security and challenges. These security and privacy issues are prevalent across all aspects of IoT ecosystem – hardware, software and communication [9][10]. Implementing security solutions in IoT is particularly complicated because they often have simple processors, with very limited memory and storage capacity [11][12].

The third major concern is related to connectivity and the device management. IoT networks are often deployed in harsh environments [13][14][15]. Unfriendly environment coupled with low power wireless links are often a major cause for degradation in a communication reliability [16]. As IoT devices are deployed in a harsh environment, the possibility of device failure is high. IoT application must be capable of monitoring and diagnosing special events or failures of various forms such as security breach and must be able to recover from a crash. Management of thousands of devices in a distributed manner requires designing an application with a high level of concurrency and it is a challenging task.

Fourth on the list of concerns is hardware and software management. Typical IoT network may contain hundreds to a few thousands of nodes. Replacing the power source, updating the firmware manually has limited scalability. However, designing a distributed and lightweight self-aware hardware and software maintenance application is a complicated task.

Choosing a wireless protocol often involves balancing many conflicting requirements such as operational cost, energy consumption, wireless range and the bandwidth. Technical trade-offs and myriad commercial solutions with no clear dominant winner make selecting wireless protocol often a tedious task and it stands as one of the concerns.

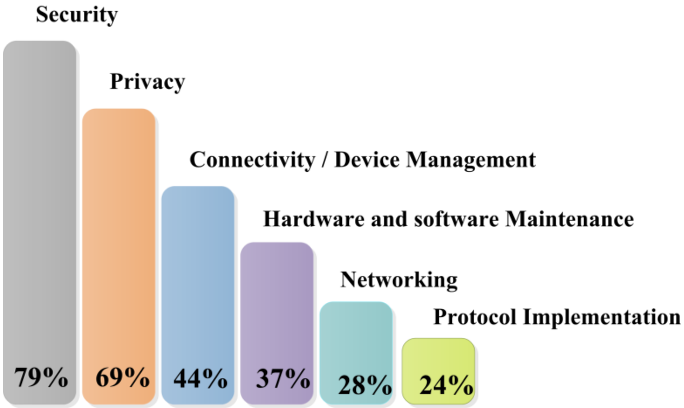


Figure 1.4 Major concerns in realizing IoT network

1.5 Device Connectivity: A Bigger Challenge

The three main tasks of any IoT network are sensing, analyzing and transmitting. Because of the communication capability, thousands of individual nodes form a single entity and provides collective intelligence on the topic of interest. Due to the limited transmission range, IoT devices may have to multi-hop their sensed data possibly over long distances. Therefore, connectivity or communication between individual devices is crucial for the successful operation of the network. Main factors that plague the reliability of the communication are discussed in the following sub-sections.

1.5.1 Internal and External Interference

An inherent problem associated with the wireless networks is interference. Interference can be internal or external. External interference is caused by other sources such as domestic appliances, Bluetooth devices. Internal interference occurs when the nodes within the same network transmit at the same time.

Wireless Sensor Networks (WSN) operate on an unlicensed ISM bands and therefore the radio spectrum is shared with several other devices [16]. As Wi-Fi uses 2.4GHz frequency, the WSN such as IoT might have to compete to utilize the frequency. Appliances such as microwave oven, plasma lighting system, cordless phones that generate electromagnetic noise are also known to increase the packet loss rate, this, in turn, increases the retransmission rates and latency

[17][16]. Similarly, internal interference due to concurrent transmission also negatively impacts the performance of the network [18]. It is also empirically shown that a cross-channel interference also decreases the packet delivery rate [19].

1.5.2 Environmental Factors

Natural obstacles such as office walls, human and vehicular movements, dense vegetation generally deteriorate the signal strength. Humidity and temperature also influences the radio waves [20][21][22][23]. The amount of signal attenuation in dBm for WLAN is documented in [24]. The communication quality of wireless devices in sports wearables are known to suffer because the human body absorb, reflects or even scatter wireless signal [25].

1.5.3 Ad-hoc Deployment

When the area to be monitored is large and in a remote location, sensors are usually randomly deployed. For instance, they are dropped from the aircraft to monitor environmental conditions such as temperature, humidity or for surveillance purpose [26]. This deployment strategy results in the nodes having different antenna orientation. The impact of antenna orientation on the performance of the sensor network is significant [27].

The position of the sink node – a data aggregation point also plays a crucial role in the performance of the network. Imbalanced traffic load causes some of the nodes in the network to expend more energy, risking the network to become unconnected and thus rendering it useless [28].

If the average number of hops between the source and the sink node is less, the latency and the energy consumption is also relatively lesser compared to the larger number of hops between source and the sink. In addition, if the number of hops between the source and the sink is less, the collision rate is less which in turn results in better packet delivery rate.

1.5.4 Software and Hardware Configuration

Application data rate, the amount of control messages generated by the routing protocol at the network layer, collision avoidance technique employed at the MAC layer greatly influences the performance of the network. Choosing the routing protocol such as Collection Tree Protocol (CTP) that generates the fairly large amount of control packets along with high application data rate increases the contention in the network [29].

Even when the transceivers of the sensor nodes in an IoT network are configured exactly, in the same way, they may distort transmitted or received a signal due to their internal noise [30][31]. The low power transmits signals are more susceptible to interference and the multi-path distortion. In addition, the remaining battery life is also known to affect the sensitivity of the transceivers [32]. Furthermore, most of the IoT devices have an in-built antenna with irregular radiation pattern. All these factors contribute to the degradation of the communication quality.

1.6 Research Objectives

Uncertainty in the propagation medium is prominent because of the issues discussed earlier. This uncertainty significantly reduces the communication reliability. Reduction in the reliability also has other effects such as higher energy consumption as nodes have to retransmit the packets.

To enhance the link reliability between pairs of IoT devices, boosting the transmission range seems to be a natural choice [33]. However, increasing the transmission power can increase the energy consumption. The majority of the IoT devices such as mobile phones, sensor nodes such as MicaZ, TelosB all operate with limited power supply. In addition, of all the hardware components of sensor nodes, transceivers consume more energy [32].

Minimizing the transmission power level (T_x) and simultaneously strengthening the communication link between a pair of nodes are conflicting goals [34]. One of the solutions to this dispute is to employ Transmission Power Control (TPC) algorithms that scale the T_x at run-time whenever the link quality is below a certain predefined threshold level. The challenge, therefore, is to design a TPC algorithm that meets following three crucial design goals

a) Minimal effect on other layers such as routing and MAC and is platform independent
b) Minimal communication overhead
c) The capability of instantly and accurately predicting the variation in the environmental condition and recommending an Optimum Transmission Power ($OptT_x$).

To devise a robust run-time TPC algorithm that fulfils the three design goals pointed out earlier, following research objectives (RO) is addressed in this thesis

- As the link quality is uncertain due to various factors, TPC algorithm may be triggered often. The frequent scaling of T_x may increase the interference or may even cause the existing communication link to break. Therefore, it is necessary to study how run-time TPC affects the performance of the multi-hop network. Hence, the research objectives (RO) is to know

  (RO1) *How does a single node performing TPC to enhance its own link quality affect the performance of the entire multi-hop network?*

  (RO2) *How do TPC influence routing and MAC protocols?*

- Although TPC algorithms primarily deal with adjusting T_x at node-level to achieve good reliability and energy efficiency, its impacts at the network level must not be side-lined. At node-level, a transmitting node employing T_x to reach the neighboring node may consume minimum energy, however, hidden-node problem emanates at the network-level causing interference and hence retransmission [35]. In addition, usage of minimum T_x translates into a weaker signal strength and may cause an unstable link at the node-level. Therefore, T_x that achieves good performance at a node-local may not necessarily be good from the whole network point-of-view. Hence, the research objectives (RO) that need attention are as follows

  (RO3) *Investigate if there is a trade-off in terms of reliability and energy consumption in adoption of LowT_x and HighT_x by the sensor nodes.*

- Instantaneous prediction of the link quality between the sensor nodes calls for the design of TPC that is proactive –the algorithm forecasts the link quality degradation before it actually occurs and increases the T_x or it finds that the existing link quality can be achieved even with lesser T_x than the current and therefore it decreases T_x. Not all predictive algorithms (e.g. machine learning) can be implemented in resource constraint devices. Therefore, a thorough assessment of various non-machine learning algorithms needs to be performed. To that extent, we address the following RO in this thesis

  (RO4) *Which other non-machine learning algorithms can predict the link variation?*

- To estimate the link quality, the majority of the TPC algorithms use network metrics such as Received Signal Strength Indicator (RSSI) that are sensitive to noise. A sensitive metrics results in the TPC algorithms changing T_x often. This may create an unstable network. Employing metrics such as Packet Delivery Ratio (PDR) makes the TPC algorithm more reactive. Therefore, in this thesis, we inquire the following question

  (RO5) *Which another alternative metric can be used by the TPC algorithms to change the* T_x*?*

- It is found that when the battery of the nodes depletes, nodes have more problem decoding received packets than to transmit packets [32]. As a result, the ACK message may not be sent by the receiving node to its sender and this force the sender to retransmit the packets and hence consuming more energy. The situation gets aggravated in proactive routing protocol that sends a substantial amount of control messages to maintain the route [36]. Knowing the battery state in advance can help the neighboring nodes to avoid those nodes whose battery are depleting faster and can avoid retransmitting the packets. In addition, individual nodes can adjust the transmission power level (T_x) based on its current battery level. Knowing the remaining energy can also be a potential input to TPC algorithm for deciding the T_x to use. Therefore, in order to prolong the lifetime of the network, it is crucial to forecast the energy usage and save the energy per transmission. Therefore, we address the following question in this thesis

  (RO6) *How accurately can machine-learning algorithms forecast the battery level of the IoT device?*

1.7 Experimental Configuration

1.7.1 IoT Device

To validate the claims and evaluate the TPC algorithms presented in this thesis, TelosB motes was used as a hardware platform [11]. As these motes are open source platform, it is a suitable device for research and experimentation of low-power IoT network. It operates with 2AA batteries and offers reliable communication capability with low power consumption. As Figure 1.5 depicts, the mote is equipped with USB connectors, low-power MCU, IEEE complaint radio with antenna, inbuilt sensors such as humidity, temperature, luminosity. It also offers the possibility of attaching additional sensors through the expansion slots. USB programming features greatly helps in reprogramming the device.

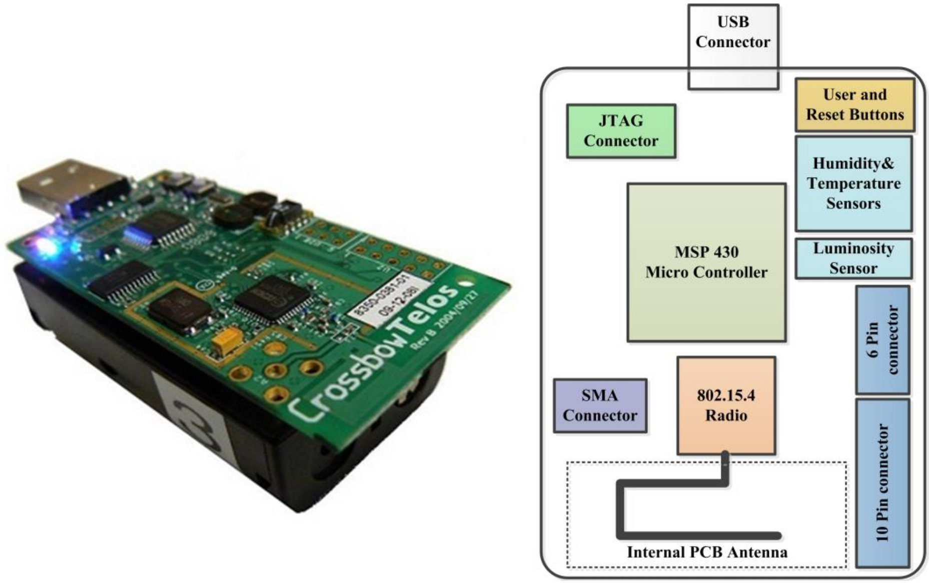


Figure 1.5 TelosB Sensor mote and its corresponding block diagram

1.7.2 IoT Operating System

The TelosB motes for our experiments run Contiki 2.7 open source operating system [37]. Contiki fully supports IPv6, IPv4, 6LoWPAN, RPL, and CoAP standards. The development of programs is easy and fast because the applications can be written in standard C programming language and therefore making it portable programs. In addition to the vibrant developer community, Table 1.1 provides other salient features that make it obvious choice compared to Tiny OS for hassle free development of complex algorithms.

Table 1.1 Features comparison of Contiki OS with Tiny OS

OS	Modularity	Real-Time	Multi-Threading	C Support
Contiki	Partial Support	Partial Support	Partial Support	Full Support
Tiny OS	No Support	No Support	Partial Support	No Support

1.7.3 Simulation Environment

For the rapid deployment and testing of a complex network, Cooja simulation environment was used [38]. The striking feature of Cooja simulator is that the network comprising TelosB can be emulated and the same code can be uploaded to the motes without any changes.

1.7.4 Networking Protocol

Although conventional cellular network has high data rates, they have high hardware and operational cost, smaller bandwidth, and higher energy consumption. The vision of IoT network is to have thousands of nodes to monitor a large area. This setting requires the node to be cheap, remain operational for many hours, and allow rapid creation of mesh network. In addition, these devices typically measure the environmental values such as temperature, humidity that are only a few bytes in size and therefore does not warrant the need for high data rates. Therefore, to represent realistic IoT scenarios, IEEE 802.15.4 protocol was used as an underlying networking protocol.

The IEEE 802.15.4 standard as shown in Figure 1.6 defines physical layer (PHY) and Medium Access Control layer (MAC) of the Open Systems Interconnection (OSI) model. The standard provides 27 radio channels in a specific unlicensed 800, 900 and 2400 MHz bands. The fading of radio frequency channel are moderated by the use of Direct-Sequence Spread Spectrum (DSSS) [39].

PHY layer defines the power, frequency and wireless link condition with a Phase-Shift-Key (PSK) transceiver capable of transmitting up to 256 kbps [39]. A typical TPC algorithm uses the values of PHY layer for adjusting the power level.

MAC layer defines the format of data handling, access to the physical channel. There are two channel access methods - Carrier Sequence Multiple Access (CSMA) with Collision Avoidance (CA), and Time Domain Multiple Access (TDMA) using synchronization beacons and Guaranteed Time Slots (GTS).

The lower layer (Layer 1 and 2) of the 802.15.4 standard communicates with the upper layer of the protocol stack through the Logical Link Control (LLC) and Service Specific Convergence Sub-Layer (SSCS). The functionality of LLC is defined in the IEEE 802.2 [39]. The SSCS adds a header or wraps the data in a header and trailer that contain necessary information such as error control and the priority information [40].

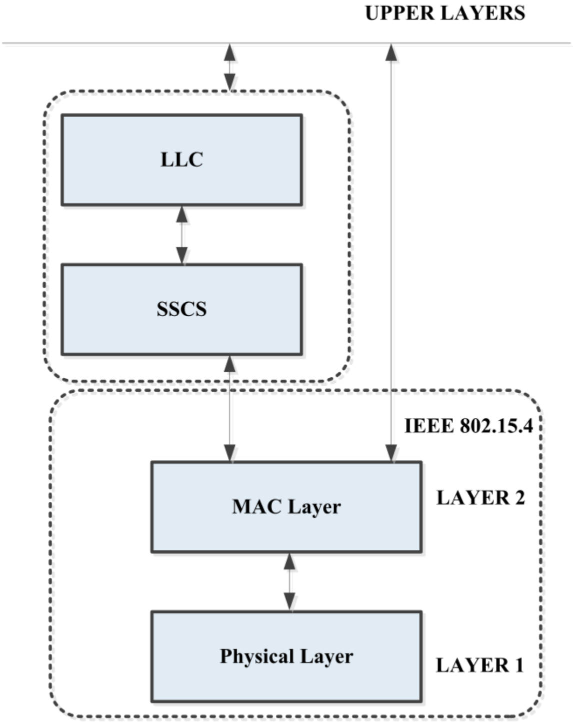


Figure 1.6 802.15.4 protocol stack

1.7.5 Routing and MAC Protocols

The main objective of TPC algorithms is to adjust the transmit power level. Therefore, it needs the information from two protocols –routing protocol for link quality estimation between devices and the channel condition information from the MAC protocols to elect optimum transmission power at run-time [35].

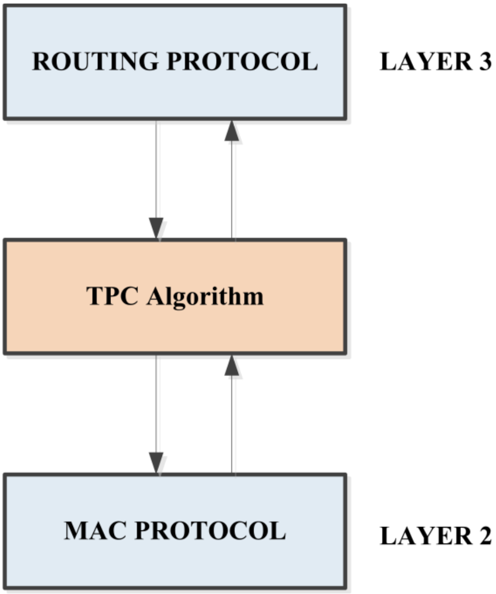


Figure 1.7 Placement of TPC algorithm in 802.15.4 protocol stack

Depending on the placement of TPC layer and its interaction with other protocols as shown in Figure 1.7, it might affect the performance of either protocol. Therefore, it is crucial to design a TPC algorithm based on the understanding of the inner working of routing and MAC protocols.

In this thesis, the proposed TPC algorithms are evaluated in the presence of three types of routing protocols -IPv6 Routing protocol for Low-Power and Lossy Networks (RPL), Collection Tree Protocol (CTP) and ContikiMesh and MAC protocol known as ContikiMAC.

1.8 Contributions and Organization of the Thesis

The main objective of this thesis is to study and design a robust TPC algorithm that enhances the reliability of the network with minimum energy consumption. Objectives of each chapter is pictorially presented in Figure 1.8

Chapter 2 presents a generic TPC framework and provides a detailed explanation of various TPC algorithms provided in the literature. It also highlights the shortcomings of the techniques of adapting the transmission power based on various parameters hardware and software metrics.

Chapter 3 shows the impact of one node performing TPC on the sizeable static network and provides the answer to research objectives *RO1* and *RO2* presented in section 1.5. This chapter also presses the need of designing TPC algorithms based on the functioning of other layers such as application, routing, and MAC. The simulation experiments show how one node's decision to scale its transmission power can affect the performance of both routing and MAC layers of multiple other nodes in the network, generating cascading packet retransmissions and forcing far too many nodes to consume more energy.

Chapter 4 studies the impact of fixed transmission power on the network and provides the answer to the research objective *RO3* presented in section 1.6. We evaluate the performance of the network of various densities in terms of packet loss, energy consumption and collision in two different scenarios – network-wide fixed minimum and maximum transmission power under fixed and random application data rate. The general misconception is that the IoT network employing maximum transmission power increases the chances of a collision resulting in higher retransmission and energy utilization. In this chapter, we show that maximum transmission power fairs well across all the metrics compared to the minimum transmission power.

Chapter 5 A detailed analyses of four potential lightweight run-time predictive algorithms for proactive TPC is evaluated in this chapter and addresses the research objective R04. Experimentation shows that linear regression algorithm has the worst prediction accuracy and exponential moving averages, weighted moving averages significantly outperform the linear regression. Discrete Kalman Filter has the highest accuracy. However, due to its implementation complexity and complex configuration, simple weighted moving averages seem to be the best algorithm for implementing the proactive TPC.

Chapter 6 presents the unique contribution in the form of proposing a proactive TPC algorithm known as Data-Aware TPC that is successful in minimizing power consumption during the initialization phase and shows that by utilizing priority of data as a sole metric for power adaptation improves reliability and decreases the energy consumption. This chapter highlights the merits of DA-TPC over the reactive TPC design and addresses the research objectives RO5 and highlighted in section 1.6.

Chapter 7 provides the information on the importance of the battery level prediction in order to achieve a connected network and tackles the research objectives *RO6* mentioned in section 1.6. This chapter evaluates the prediction accuracy of determining battery depletion rate of various machine-learning algorithms. The experimentation shows that the nature of the network metric values does not allow for an early and accurate prediction of the stationary node's current voltage level and it further stresses the need for employing efficient TPC algorithms.

Chapter 8 draws the conclusion and discusses future research direction and problems.

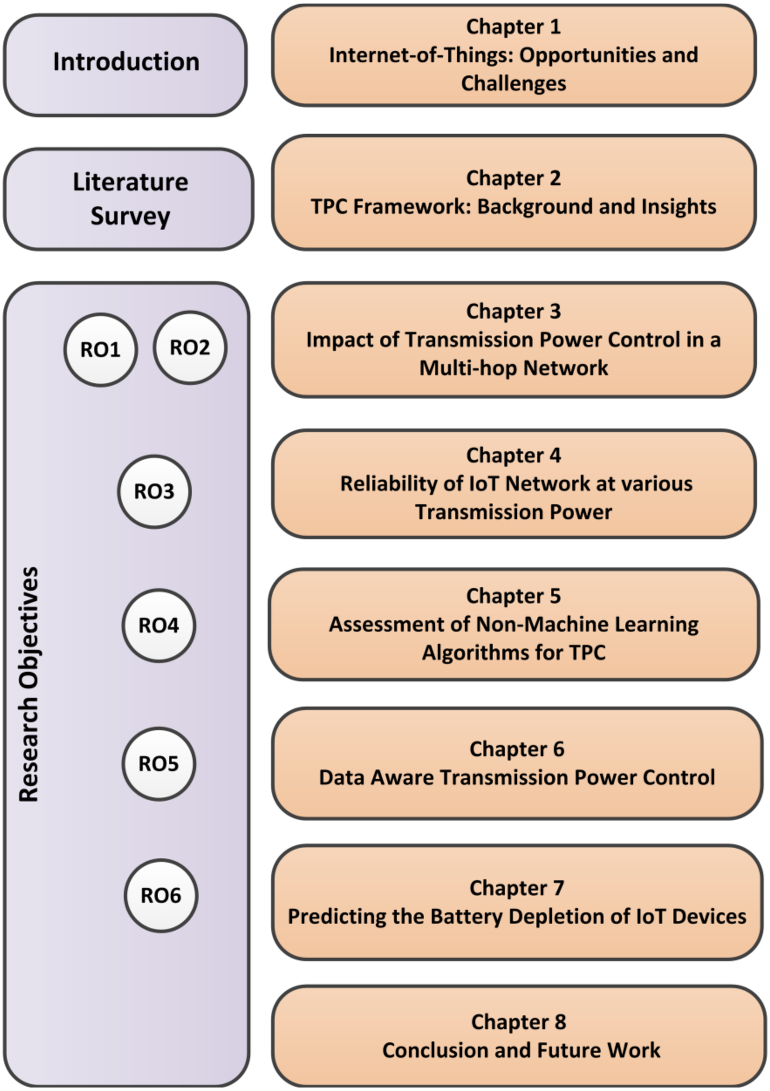

Introduction
Chapter 1
Internet-of-Things: Opportunities and Challenges
Literature Survey
Chapter 2
TPC Framework: Background and Insights
Research Objectives
RO1
RO2
Chapter 3
Impact of Transmission Power Control in a Multi-hop Network
RO3
Chapter 4
Reliability of IoT Network at various Transmission Power
RO4
Chapter 5
Assessment of Non-Machine Learning Algorithms for TPC
RO5
Chapter 6
Data Aware Transmission Power Control
RO6
Chapter 7
Predicting the Battery Depletion of IoT Devices
Chapter 8
Conclusion and Future Work

Figure 1.8 Research objectives addressed in each chapter

Chapter 2 TPC Framework: Background and Insights

Limited energy and interference are the two unique problems affecting the performance of a wireless network. The uncertainty in the channel condition significantly reduces the communication reliability. Fall in the reliability also has other adverse effects such as reduction in the lifetime of the node as packets may have to be retransmitted. Drastic variations of link quality in indoor and outdoor IoT deployments has for long motivated the need for Transmission Power Control (TPC) techniques. A TPC algorithm or framework that adjusts the transmission power to ensure good communication link quality with minimum energy expenditure can be achieved only when it fulfils the three essential design goals outlined in chapter 1. Thus, a thorough understanding of the TPC presented in the literature is needed, which is the purpose of this chapter.

The remainder of this chapter is as follows. In section 2.1, a general framework of TPC that consists of three main components is discussed. Taxonomy of various metrics utilized for estimating the link quality between a pair of wireless devices is presented in section 2.2. Section 2.3 discusses diverse algorithms that process the metric values and recommends the transceiver to adjust the transmission power (T_x) at run-time. Section 2.4 outlines the general drawbacks of the TPC frameworks. Lastly, section 2.5 presents the conclusions.

2.1 General Framework of TPC

The three fundamental components that are necessary for adjusting T_x are shown in Figure 2.1[16]. From the top, the framework consists of a component that monitors the link quality metrics for a certain time between a pair of wireless devices. Monitoring a link can be classified as active, passive or hybrid link monitoring [16]. In active monitoring, additional probe packets other than the normal data are transmitted at a certain rate to deduce link trait [41]. In the passive case, on-going data communication between wireless devices is used to infer the link quality without additional communication overhead [42]. However, a node listens to the packets, even if they are not addressed to them. As the node overhears, the transceiver is in the listen state consume more energy [43]. Lastly, in hybrid monitoring, as the name suggest, both aspects of previous two strategies are combined to obtain up-to-date link analysis [44].

The second component from the top is responsible for retrieving the link quality metrics either from the sent packets or from the received packets. Hence, depending on where the metric is retrieved, one can have a source-side or a receiver-side TPC algorithm.

The last component of the framework evaluates the metrics and recommends the T_x adjustment needed to obtain node-level reliability. The evaluation algorithm can be a simple moving average that computes the metric for a certain time window or a complex machine learning algorithms such as neural network [45][46].

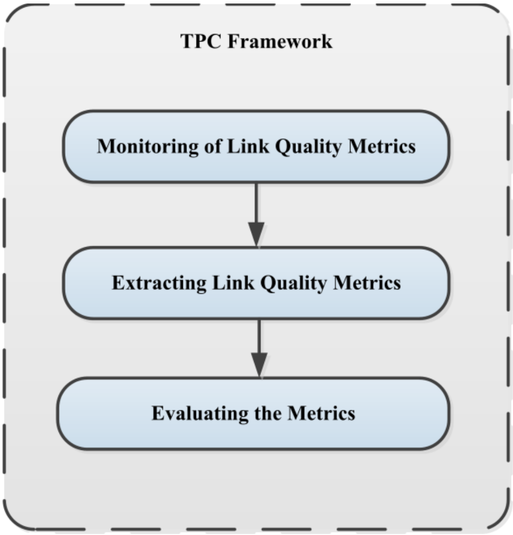

TPC Framework
Monitoring of Link Quality Metrics
Extracting Link Quality Metrics
Evaluating the Metrics

Figure 2.1 TPC Framework[1]

2.2 Classification of Link Quality Metrics

Figure 2.2 shows some of the main link quality metrics. There are two broad categories of link quality metrics – hardware and software.

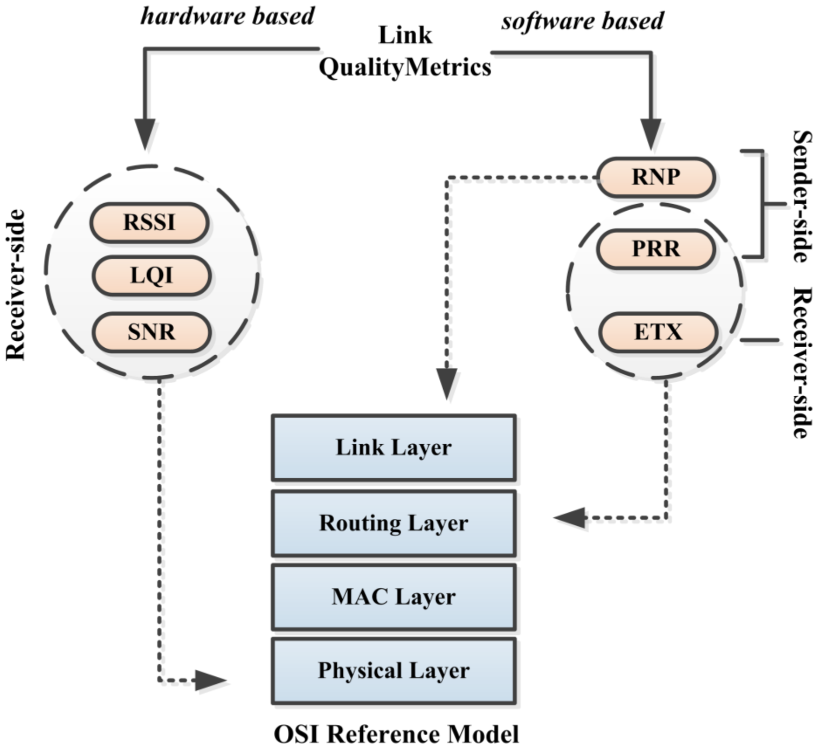


Figure 2.2 Various link quality metrics from various OSI layers

2.2.1 Hardware Based Metrics

Metrics such as Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), and Signal-to-Noise (SNR) can be obtained directly from the transceiver of the sensor nodes and are retrieved from the physical layer - hence they are categorized as hardware based link quality metrics. Hardware based metrics are embedded in the incoming data packet and hence they can be easily extracted.

According to the 802.15.4 standard, the physical layer must provide two metrics –RSSI and Link Quality Indicator (LQI) for all the received packets [47]. RSSI measures the strength of the received packet over eight symbols. It is measured in dBm and ranges from -100 dBm to 0 dBm [48].

Similarly, LQI measures the chip error rate over the 8-bit period after the start of frame delimiter (SFD). LQI values are between 110 and 50. An LQI value represents the average symbol correlation value over the packet's first 8 symbols [48]. The packet format of LQI and RSSI is shown in Figure 2.3.

Lastly, SNR (Signal-to-Noise Ratio) is computed as the difference in decibel between the received signal and the background noise level. The possibility of data corruption and packet retransmission is higher if the received signal is closer to the noise level [16].

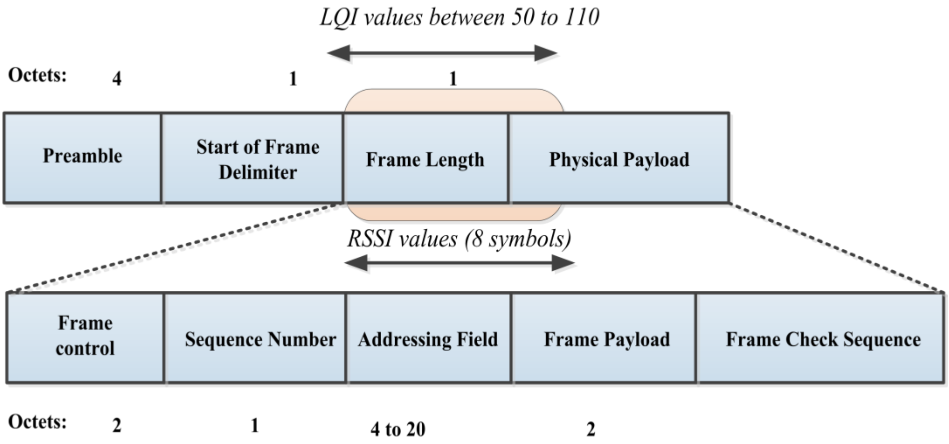


Figure 2.3 Packet format of LQI and RSSI [9][10]

2.2.2 Software Based Metrics

Software based metrics require additional computation to be derived. For example, the metric Required Number of Packet Transmission (RNP) must count the number of transmitted and retransmitted packets during a specific time interval *t* divided by the number of successfully received packet minus one [42]. It is a sender-side metric and assumes a node will repeat the transmission of a packet until it is correctly received [16].

Similarly, Packet Reception Rate (PRR), also known as Packet Delivery Ratio (PDR) is a sender-side metrics computed as the total number of packets successfully received divided by the total number of packets sent [49]. The value of PRR varies between zero and one. Some of the other variants of PRR are Packet Acknowledgement Rate (PARR) and Packet Error Rate (PER). The latter is calculated as 1- PRR and the former is calculated as the ratio of a number of acknowledged packets to the total number of transmitted packets in a specific time window [50][16].

Finally, the Expected Transmission Count (ETX) is a receiver-side metric computed as the inverse of the product of the Acknowledgement Reception Ratio (ARR) and the PRR [16][51]. The value of ETX varies between one and infinity. Some of the ETX variants are mETX (Modified ETX) [52]. ETX does not perform well under short-term channel variations. This is because it employs mean loss ratio for making a routing decision. To address this mETX is proposed[53].

2.3 Classification of TPC Algorithms

As shown in Figure 2.4, we broadly classify the algorithms that evaluate the link quality metrics into non-machine learning and machine learning approaches.

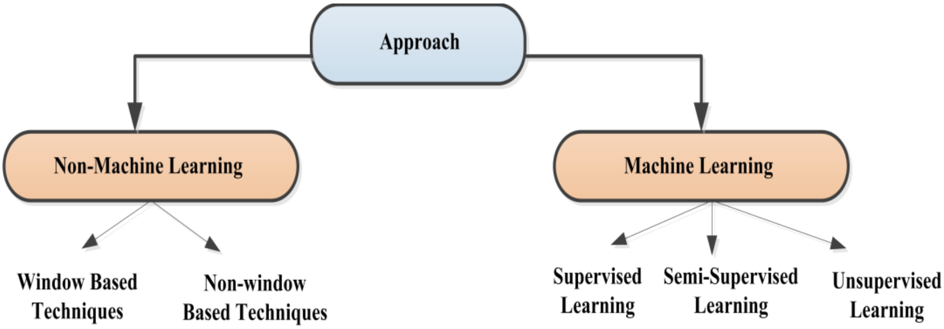


Figure 2.4 Various approaches to evaluate the link quality metrics

2.3.1 Non-Machine Learning Approaches

As depicted in Figure 2.4, this approach broadly employs window and non-window based techniques. The concept of the window-based technique is explained in equation 2.1 and 2.2 Software based Link Quality Metrics (LQM) such as PRR is collected and is averaged over a specific time window *w*. If the averaged Link Quality (*LQ*) value is below or above a predefined threshold *θ*, the transmission power is scaled up or down respectively.

$$Link\ Quality(LQ) = \frac{\sum_{i=1}^{w} LQM}{w} \tag{2.1}$$

$$\left.\begin{matrix} LQ \le \theta \\ LQ \ge \theta \end{matrix}\right\} TPC = TPC \pm 1 \tag{2.2}$$

As the packets are averaged over a period, the window-based approach is reactive to the change in the link quality. Few window based TPC algorithms correlate PRR with RSSI, LQI, or SNR. Table 2.1 presents some of the window based TPC algorithms.

Table 2.1 Non-machine learning TPC algorithms that employ software based metrics and window based techniques

Group 1: Window based TPC		
Software Based Metrics	**Non-machine learning TPC algorithms**	**Correlation**
PRR	1. Adaptive and Robust Topology Control (ART) [33] 2. Transmission Power Control with Black listing (PCBL) [54] 3. Practical TPC (P-TPC) [55] 4. Kalman Filter based Link Quality Estimation [56]	None None None SNR

In the non-window based technique, historical values of the metrics are not stored to calculate the link quality, instead, the link quality is calculated instantaneously based on the previous value of the metric, and desired transmission power level is chosen that adheres to the predefined threshold level for every future transmission of the data packet. This is explained in the equation 2.3 and 2.4 Table 2.2 presents some of the non-window based TPC algorithms and the metrics utilized by them.

$$Link\ Quality\ (LQ) = RSSI_{pkt-1} \tag{2.3}$$

$$\left.\begin{array}{l} RSSI_{pkt-1} \le \theta \\ RSSI_{pkt-1} \ge \theta \end{array}\right\} TPC = TPC \pm 1 \tag{2.4}$$

Table 2.2 Non-machine learning TPC algorithms that employ hardware based metrics and non-window based technique

Group 2: Non-window based TPC	
Hardware Based Metrics	**Non-machine learning algorithms**
RSSI	1. On demand Dynamic TPC (ODTPC) [57] 2. Adaptive On demand Dynamic TPC (AODTPC) [58] 3. TPC based on Binary Search (TPC-BS) [59]

2.3.2 Machine Learning Approaches

TPC algorithms that utilize learning algorithms to evaluate link quality metrics are more predictive than their counterparts are. Figure 2.4 shows that the machine learning algorithms are classified into three groups- supervised, semi-supervised and unsupervised.

Supervised learning algorithms analyze the externally labelled data and construct an inference function that can be used to classify future data samples. The supervised learning scheme has two phases- training and classification. In the training phase, the supervised learner is trained by providing the labelled data set. In the classification phase, the learning algorithm is given the data without the label and the algorithm must be able to classify the data.

Semi-supervised learning algorithms are similar to supervised learner programs. However, during the training phase, they are provided with a small amount of labelled data. One of the motivations behind using a semi-supervised learner is intrinsically linked to the improvements in the performance of a computation model. It also reduces the process of manual labelling of data sets [60].

Unlike its counterparts, an unsupervised learning algorithm, without any externally labelled data classifies the data input. Hence, it does not have a training phase. These algorithms expect the observed data to have some analytical forms. As link quality metrics such as RSSI, LQI vary abruptly, they are not a valid input to the unsupervised learner and hence they are not frequently used to evaluate the link quality and recommend changes in the T_x level. Table 2.3 provides a partial list of various TPC algorithms that employ machine-learning algorithms.

Machine learning can also be classified as offline or online learning. In the former process, the entire data is processed at once before generating the predictor (e.g. algorithms 1, 3, 5 of Table 2.2). In the latter process, a predictor is generated as and when new data arrives sequentially (e.g. algorithms 2, 4, 6 of Table 2.2)

Table 2.3 TPC algorithms that employ various machine-learning algorithms

TPC Algorithms	Learning Algorithms	Learning Type	Metric	Metric Type	Correlation
1. Predicting link quality using supervised learning in WSN [61]	Decision Tree	Supervised	LQI	Hardware based	None
2. Online supervised learning of Link quality estimates in wireless network [62]	Locally Weighted Projection Regression	Supervised	PRR	Hardware and Software based	RSSI, SNR
3. Bio-inspired link quality estimation for wireless mesh networks [45]	Neural Network	Supervised	ETX	Software Based	None
4. Foresee (4C): Wireless Link Prediction using Link Features [63]	Logistic Regression, Neural Network	Supervised	PRR	Hardware, Software Based	RSSI, LQI and SNR
5. Link Quality Assessment Model for WSN [64]	Support Vector Machine	Supervised	PRR	Hardware, Software Based	RSSI
6.Reinforcement learning in power control games for internetwork interference mitigation in Wireless Body Area Networks [65]	Reinforcement learning	Semi-supervised	SNR	Hardware Based	None

2.4 Limitations of TPC Algorithms

In this section, we first present the drawbacks of various link quality metrics and then highlight the limitations of various algorithms presented in Figure 2.4.

2.4.1 Drawbacks of Hardware based Link Quality Metrics

RSSI measurement is extensively used in numerous sensor network protocols such link estimation [66][67], TPC [57][68] and distance estimation [69][70]. Many of these protocols use the raw RSSI measurement that the radio provides which has an accuracy of ±6 dBm [71]. ±6 dBm is a wide error margin and it is shown that with a small variation (2 to 3 dBm) in RSSI, the Packet Reception Rate (PRR) can decrease from 100% to 0% [72]. It is shown that sensor nodes with 802.15.4 compliant radios such as CC2420 [73] and Atmel AT86RF230 [74] introduce systematic errors in their RSSI measurements [72]. In addition, the existence of non-linearity between the transmit power and the corresponding RSSI values and its profound impact on the network is presented in [72]. Furthermore, the fluctuation in RSSI is prominent, causing the TPC algorithms to change the T_x level [75]. This change in the power level increases the interference and, in turn, can reduce the reliability of the network.

The LQI (Link Quality Indicator) metric represents the average symbol correlation value over the packet's first eight symbols. Similar to RSSI, experiments show that LQI is not a good indicator for estimating the intermediate quality links due high variance [33]. LQI must be averaged over many samples before it can really estimate the link quality [16].

To compute SNR, one must calculate the difference between the signal RSSI (RSSI corresponding to the ambient channel noise) and noise RSSI (RSSI corresponding to the received packets) [72]. Therefore, the computation of SNR indicates that the transceiver must be kept in listen mode for prolonged periods of time. Popular transceivers such as CC2420 consume more energy in the listen-mode than in transmitting mode [73]. A widely used operating system for IoT such as Contiki uses ContikiMAC as the MAC protocol that keeps the transceiver in sleep mode 99% of the time [37][76]. This calls for the reconfiguration of the MAC protocol that may result in undesired results. In addition, calculation of SNR is problematic when there are sensor nodes that transmit packets simultaneously.

Because of the shortcomings of hardware based metrics, single metrics do not provide the much needed insights into the quality of the link. Therefore, the mapping between RSSI and PRR, LQI and PRR and SNR and PRR is performed. However, the correlation between them is still much of a debate in the sensor network community. For example, it is shown that correlation between RSSI and PRR, LQI and PRR and SNR and PRR is not accurate [16][33], while in other research work it is shown that both RSSI and LQI can accurately predict the PRR [68]. Similarly, the relationship between PRR and SNR are found to be noisy and unpredictable [72].

2.4.2 Drawbacks of Software based Link Quality Metrics

As PRR is an unbiased metric, to evaluate the accuracy of hardware based estimators, it is widely used in routing protocols [77][78]. However, the link reliability depends on the number of PRR values evaluated. It is experimentally shown that for links with very high PRR or low PRR, the accuracy of estimating link quality can be achieved within a shorter time. However, the links with medium PRR value require larger time window to estimate the link quality [79]. PRR-based metrics assumes that the current link quality remains the same as the last estimation. This assumption of stable link quality is incorrect due to frequent variations of wireless links [80].

To overcome the disadvantage of minimum hop count, the ETX metric was proposed. Although ETX is a comparatively better metric for estimating the link quality, it is not immune to common problems prevailing in sensor networks. For instance, ETX and its variants (mETX) show poor performance under high traffic load [81][16][82].

PRR is a receiver-side link estimator and RNP is a sender-side estimator. This shows that when the quality of the link is poor, packets cannot be received and hence PRR cannot be computed. However, RNP can be computed even if the packets do not arrive at the destination. Therefore, RNP can underestimate the quality of the link as packets are retransmitted many times before being successfully received. Counting based metrics such as RNP or PRR cannot adapt to changes in the wireless channel fast enough [83][16].

2.4.3 Drawbacks of TPC Algorithms

Based on the working of various TPC algorithms proposed in the literature (refer to Table 2.1-2.3) and the metrics they employ, Figure 2.5 highlights the three main classes of drawbacks present in them. Each main demerits is explained in respective sections with corresponding numbers.

Data Issues

1. As we discussed in sub-sections 2.4.1 and 2.4.2, hardware based metrics- RSSI, LQI varies a lot. Frequent and abrupt fluctuations in their values create many outliers in the data points. Providing learning algorithms with sharp variations can give a less accurate prediction on the link quality [84].

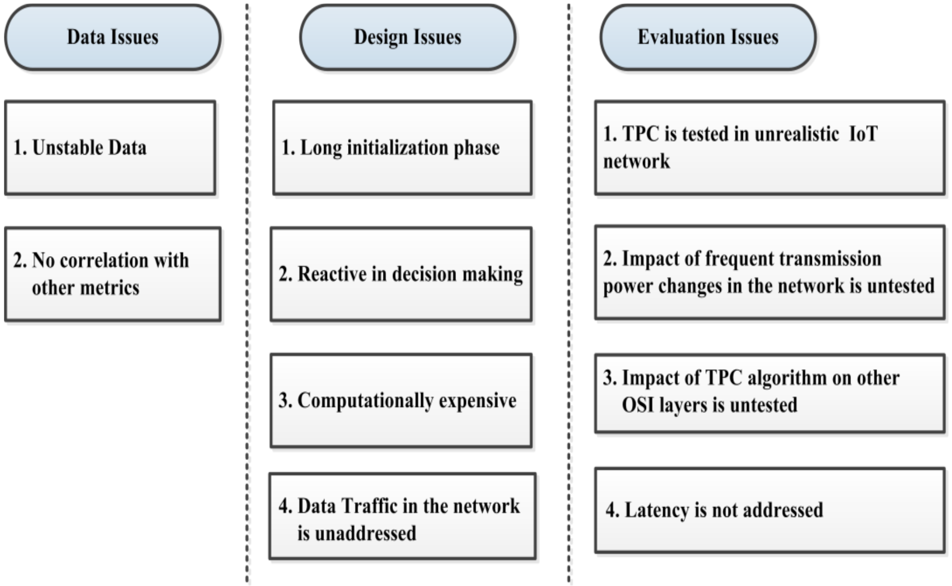

Data Issues
1. Unstable Data
2. No correlation with other metrics
Design Issues
1. Long initialization phase
2. Reactive in decision making
3. Computationally expensive
4. Data Traffic in the network is unaddressed
Evaluation Issues
1. TPC is tested in unrealistic IoT network
2. Impact of frequent transmission power changes in the network is untested
3. Impact of TPC algorithm on other OSI layers is untested
4. Latency is not addressed

Figure 2.5 Drawbacks of TPC algorithms

2. Although hardware link quality metrics provide quick updates on the link quality, PRR is a more appropriate and direct metric that can determine the reliability of the connection. Therefore, many of the TPC algorithms correlate the hardware metrics with PRR. However, this correlation is a subject of debate in the research community and the interpretation of it has always been equivocal. For example, it is shown that RSSI has good correlation with PRR in some indoor and outdoor locations [67][68]. While other research indicates quite opposite results [33][54].

Design Issues

1. To investigate the temporal changes in the link quality, many TPC algorithms listed in Table 2.1 and 2.2 have a long initialization phase (init phase). For example, PCBL, ATPC algorithms transmit a large number of probe packets at different T_x levels. When the network is dense and has a requirement of becoming operational with smaller up-time duration, the init phase is often an overhead that consumes additional energy.

2. The TPC algorithms presented in the Table 2.1-2.2 are reactive in their decision-making. Only after the link quality falls below a certain threshold level for a certain time window w, the algorithm tries to enhance the link quality and it does not predict the future link quality level.

3. In order to predict the link quality level, machine learning algorithms (refer Table 2.3) are employed. However, this requires that algorithms be trained before they are deployed. Hence, they have a long init phase. In addition, not all learning algorithms can be deployed in these devices due to hardware constraints of many IoT devices such as TelosB, MicaZ etc. The computational cost of various learning algorithms is provided in [85]. The conventional approach to supervised learning is to first obtain a lot of random samples of training data and then to label it before any learning commences. To minimize the cost of labelling, online learning is used. However, to obtain superior results from the online learning, they must employ the concept drift mechanism. The absence of this results in inferior prediction accuracy [86] In addition, as online learning makes a real-time decision as and when the data arrives, it may introduce frequent fluctuation of transmission power level. The drawback of this in the network will be discussed shortly (see Evaluation issues).

4. The rate of application data (e.g. temperature) generated and the amount of link quality metric value required to estimate the link quality also needs to be considered when designing the TPC algorithm. Imagine a source node transmitting the application data every 5 seconds and TPC algorithm running in the receiver node requires 10 PRR values to decide if the link quality is good or bad. This indicates that receiver node recommends the source node to change the T_x level only after 50 seconds though the link quality has degraded long back. Experimenting to find a balance between the application data rate and the number of LQM values required to successfully understand the link variation is important for early detection of link failure. However, different data rate and the number of LQM needed to correctly evaluate the link are not tested in many TPC algorithms mentioned in the literature.

Evaluation Issues

1. The majority of the TPC algorithms mentioned in Table 2.1-2.3 are tested in a single hop network. The TPC algorithms running in the nodes are tested in a round-robin fashion where only one node transmits a certain amount of packets at a certain power level. While rest of the nodes are listening and they recommend an appropriate power level as and when the link quality varies. A large IoT network does not operate in a round-robin fashion. A real-world network rather has many multiple source nodes transmitting the data simultaneously, possibly along with several hops. When this is the case, the instability in the input data (link quality metrics) further increases. The robustness of the TPC algorithm must be evaluated in a high traffic condition. Failure to do so provides a partial view of its performance.

2. The drawback of TPC algorithms (Table 2.1 and Table 2.2) that has long initialization phase is outlined in sub section “design issues.” To overcome this disadvantage, per-packet based TPC algorithms are proposed (Table 2.2 and algorithms 2 and 6 of Table 2.3). However, the analysis is not presented in the literature that shows the impact of frequent adaptation of transmission power level on the communication reliability of the neighboring nodes. Although TPC is used to strengthen the communication link between a pair of nodes (local level), the effect of it must be evaluated at the network level (global level) as well.

3. The performance of TPC should not be evaluated in isolation. The effect of frequent T_x scaling on the MAC and routing protocols also must be tested. As the T_x level is raised, the MAC protocol detects the collision and defers the communication causing higher latency and retransmission leading to an uptick in energy consumption. Similarly, the wrong usage of power level may force the routing protocol to select the inefficient path–longest path, high interference path etc. In the literature, the impact of TPC algorithms on other layers is not addressed.

4. Smaller transmission power directly results in shorter transmission range. This translates to the comparatively larger number of hops the packet has to make before it reaches its destination. This increases the latency in the larger network. In addition, a smaller number of hops will give rise to hidden node terminal and may cause more contention in the network than higher transmission power. Comparative analysis of selecting lower transmission power and thereby increasing the number of hops versus selecting higher transmission power and reducing the hops is not performed in TPC algorithms mentioned in the Tables 2.1-2.3. This study is needed to estimate the benefits of dynamic TPC over static TPC.

2.5 Conclusions

In this chapter, we provided the description of generic TPC framework. Details regarding classification of link quality metrics, their explanations, and drawbacks were outlined. Brief working of potential algorithms (machine learning and non-machine learning) from the literature that evaluates the LQM are presented along with their drawbacks. In the coming chapters, we address the research objectives presented in chapter 1. Future chapters also show how three main issues related to the data, design and evaluation methodology when not addressed properly could influence the reliability of the network.

Chapter 3 Impact of TPC in Multi-Hop Networks

Many Transmission Power Control (TPC) algorithms have been proposed in the past, yet the conditions under which they are evaluated do not always reflect typical Internet-of-Things (IoT) scenarios. IoT networks consist of several source nodes transmitting data simultaneously, possibly along multiple hops. Link failures are highly frequent causing the TPC algorithm to kick-in quite often.

Frequent fluctuation of the transmission power (T_x) in a dense network consisting of multiple source nodes increases the degree of interference resulting in a collision and ultimately impacts the latency and PDR. Most of the TPC algorithms are not tested in a multi-hop scenario. The situation of collision is magnified when the underlying routing protocols such as Collection Tree Protocol (CTP) [36] periodically transmit control messages to keep the network connected. Medium Access Control (MAC) protocols such as ContikiMAC [76] are designed to retransmit the packets (control packets from routing protocol and data packets such as temperature or humidity from the application layer) until they receive the ACK from the receiving nodes. Needless to say, retransmission obviously increases the energy consumption.

To this end, in this chapter, we address the following research objectives mentioned in chapter 1

(RO1) *How does a single node performing TPC to enhance its own link quality affect the performance of the entire multi-hop network?*

(RO2) *How do TPC influence routing and MAC protocols?*

To tackle these research objectives, experiments were carried out in a popular simulation environment known as COOJA [38] and we extensively study and examine the impact of one node performing TPC frequently for a short period of time on a relatively dense and sizeable static network consisting of 10 nodes.

The performance of the network is analysed in terms of Packet Delivery Ratio (PDR), Latency, Energy Consumption at two different levels of the network namely global-level and at the node-level. The nodes use MAC protocol known as ContikiMAC and two different routing protocols – ContikiMesh and CTP separately.

The experiments show how one node's decision to scale its transmission power can affect the performance of both routing and the MAC layers of multiple other nodes in the network, generating cascading packet retransmissions and forcing far too many nodes to consume more energy. We find that crucial objectives of TPC such as conserving energy and increasing the network reliability are severely undermined in multi-hop networks.

The remainder of the chapter is organized as follows. In section 3.1, we highlight the shortcomings of the analysis under which popular TPC algorithms are evaluated. Section 3.2 briefly describes two different routing protocols – ContikiMesh, CTP, and ContikiMAC. Section 3.3 provides the information about simulation parameters and the experimental setup. Section 3.4 discusses and summarizes the results. Conclusions are provided in section 3.6.

3.1 Related Work

In this section, we explain how TPC can impact the normal operation of MAC and routing layers. We highlight the drawbacks of some of the known TPC algorithms provided in Table 3.2

3.1.1 General Limitations

Several TPC algorithms shown in Table 3.2 have been proposed in the literature. However, the majority of them are tested in a single-hop network. Furthermore, these algorithms running in nodes are tested in a round-robin fashion where only one node transmits a certain amount of packets at a certain transmission power level (T_x) and the rest of the nodes are listening. Finally, the inference is drawn on per link basis.

A realistic IoT network does not operate in a round-robin fashion, low contention environment. A real-world network rather has many multiple source nodes transmitting data simultaneously possibly along with several hops. Imagine an IoT network deployed in a vineyard that monitors humidity, temperature and transmits the data along a multi-hop to the aggregation unit [87]. Large geographical areas leave the resource constraint nodes no choice but to multi-hop its data even with its maximum transmission range (e.g.: CC2420 has a maximum of 50m transmission range). Thus, avoiding routing protocol is not possible.

When there is a dense network, without a MAC protocol performing CSMA/CA, there is going to be a lot of packet drops due to the collision. It is already a known fact that a node in the listening or idle state still consumes more energy than in transmit mode [88]. Therefore, a MAC protocol with the integrated sleep/wakeup capability reduces the listening/idle time of the node.

To have a large distributed and yet efficient IoT network, one has to reap the benefits provided by the routing and MAC layers. On the contrary, the induction of TPC in a dense multi-hop network can flip the merits of a MAC and route layer into demerits. As TPC resides between the routing and MAC layer, it's impact on other layers or other layers impact on the performance of TPC has to be thoroughly examined in a realistic scenario (multi-hop network with multiple source nodes) [35]. Therefore, the performance of TPC algorithms cannot be evaluated in isolation.

3.1.2 Implications of TPC on MAC Layer

MAC protocols help a pair of nodes to synchronize their communication. This aids the transmitting node to transmit data at the time interval when its recipient is in the listen mode. The transmitting node transmits to its recipient based on the communication pattern of its neighbor. Under fixed T_x, the MAC's communication sync strategy works well in avoiding a collision and mitigates the hidden terminal problem but is known to aggravate the exposed terminal issue [11]. In a dynamic environment as the link quality between any pair of nodes falls, a boost in T_x is triggered by TPC algorithm running in either of the nodes to compensate the error. This raise in T_x disrupts the communication sync of another neighboring pair of nodes. As a result, other nodes in the vicinity may retransmit with high T_x several times than it normally does before it is successfully received by its recipient. This may further increase the contention. Hence, the network exhibits a vicious behavior and one can see degradation in the overall performance, as discussed in [11]. Therefore, the design of TPC algorithms must take the functioning of MAC protocols into account. An overview of how a generic TPC algorithm may turn the good feature of MAC protocol – ContkiMAC against the network is presented in section 3.3.

3.1.3 Implications of TPC on Routing Layer

Normally, the user of the IoT network determines the reliability and the reduction in the energy cost to be achieved. Typical TPC translates this to optimum T_x level by correlating RSSI and/or LQI, SNR to the PDR. Unfortunately, these correlations do not yield good results due to the sensitivity of RSSI and LQI [33]. This kind of translation without taking the topology of the network into consideration might result in the node not converging to appropriate T_x level. A wrong usage of the power level would force the routing protocol to choose the inefficient path – longest path, less energy efficient path [89], high interference path [90] or high Expected Transmission Path (ETX) [91].

Many of the routing protocols generate control messages for the maintenance or fault tolerance purposes. For example, the performance of various routing protocol such as RPL, AODV, CTP and DSR in terms of PDR, latency, the number of control messages, power consumption and fault tolerance in the fire emergency scenario is discussed in [92]. The performance of the fire emergency scenario is evaluated for a static network and fixed transmission power. Table 3.1 provides the partial results of that test case.

Similarly, the performance of routing protocols such as Optimized Link State Routing (OLSR) and Dynamic Manet On-Demand (DYMO) in realistic urban test-case is discussed [93]. The introduction of TPC in both the test-cases can increase the contention and may raise the value of the metrics (latency, control messages, battery consumption) and this is not desired.

Table 3.1 Performance metrics of various routing protocol derived from [11]

Metrics	CTP	AODV
Delivery Ratio (%)	96.98	100
Average Latency (secs)	12.21	9.30
No. of Control messages	1508	3163
Power consumption (% battery remaining)	58.09	53.32
Fault Tolerance (secs)	199	78

A node that scales down the power may cause its already established link to break. Scaling up the power by the node may cause the same problem but in the neighbouring nodes due to contention. Depending on how adaptive the routing protocols are, this link breakage will be identified. To repair the link, routing algorithms will spit out extra control messages causing higher contention and energy consumption. An overview of how a generic TPC algorithm may turn the good feature of routing protocols – ContkiMesh and CTP against the network is presented in section 3.2

3.1.4 Limitations of Popular TPC

Keeping the implications of TPC on other layers (routing and MAC), Table 3.2 presents the shortcomings of TPC proposed in the literature.

Table 3.2 Design choice considered in various TPC algorithms

**Convergence (Conv), Aggregation (Aggr) and Point-to-Point(P2P)*

TPC Algorithm	MAC Protocol	Routing Protocol	Multi-hop Network	Multiple nodes Transmitting	Data Rate	Node Deployment	Traffic Pattern
ART [33]	TinyOS Default CSMA/CA	CTP	yes	yes	200 packets in 30 mins	10 nodes placed randomly. Multiple source nodes and 1 sink	Conv
DTPC [29]	B-MAC	Mint Route	yes	no	1 packet per 2 secs	22 nodes placed uniformly. Many source nodes and 1 sink	Conv and Aggr
RPAR [94]	B-MAC	RPAR	yes	yes	1 packet every 300 ms	130 nodes placed randomly. Multiple source nodes and 1 sink	Aggr
MPC [95]	TinyOS Default CSMA/CA	no	no	yes	1 packet every 3 secs	6nodes placed randomly. Multiple source nodes and 1 sink	Conv
ATPC [68]	TinyOS Default CSMA/CA	No information	yes	No	15 packets per secs	43 nodes placed randomly. Multiple source nodes and 1 sink	Conv
P-TPC [55]	TinyOS Default CSMA/CA	No	no	no	20 packets per mins	24 nodes placed randomly. Single source nodes	P2P
ODTPC [57]	B-MAC	AODV	yes	no	1 packet every 5 secs	7 nodes placed randomly. 1 source node	Aggr

From Table 3.2, one can notice that only 3 (ART, RPAR, and MPC) out of 7 TPC algorithms were tested in a multi-hop network with multiple source nodes sending data simultaneously. However, MPC and ART are tested in a network with convergence traffic. This means that the source nodes are within the transmission range of the sink forming a kind of simple star topology. A real world network would normally have a sink that is beyond the transmission range of the majority of nodes thereby forcing the traffic to be aggregated at a certain point –sink.

Furthermore, a TPC in a network with aggregated network traffic has no noticeable performance improvements over fixed transmission power scheme as shown in [29]. Therefore, it is necessary to study extensively the effect of generic TPC model in a multi-hop network with different routing protocols and aggregated network traffic.

3.2 Background

In this section, we briefly explain the inner working of two routing protocols namely ContikiMesh and CTP and a MAC protocol, ContikiMAC.

3.2.1 Generic Transmission Power Control Model

The decision to scale the power either way by TPC algorithms is obtained by the link quality metrics such as Link Quality Indicator (LQI) and Received Signal Strength Indicator (RSSI). These metrics are provided by the transceivers such as CC2420 of TelosB motes. RSSI is available in all the incoming packets and can be easily extracted. A typical approach to transmission power scaling by TPC algorithms is as follows. The receiving node calculates the RSSI from the incoming packets of the source node and then analyses if it is within the thresholds. This threshold value depends on a specific TPC algorithm. For example, MPC TPC sets RSSI threshold level to -55dBm [95]. As RSSI value can be correlated to the transmission power, a fall below the lower bound or above the higher bound of threshold immediately triggers the receiving node (R_n) in sending back an ACK packet to the source node (S_n). This ACK packet contains the recommended transmit power (T_x) for that specific S_n node that it must use for transmitting future data packets. This technique as seen in AODTPC [58] ensures that the fading link quality is quickly rectified by raising or lowering the transmit power to compensate the variation.

Other generic methods, found in P-TPC [55] and ART [33], calculate the PDR or the number of failed transmissions (determined by not receiving ACK for the packet sent) for a given window w. If either one falls below or above a respective threshold, appropriate T_x level is selected for the future transmission of packets.

Although the metrics such as LQI and RSSI are inexpensive, they are sensitive to environmental disturbances and the probability of them deviating from the predefined threshold level remains high [96]. This causes the node to change the T_x quite often and might fail to converge sooner to an appropriate power level. In addition, the R_n must inform the S_n of its RSSI or LQI quality. This backward ACK communication consumes additional energy.

Algorithms such as P-TPC and ART, on the other hand, do not rely on sensitive RSSI, LQI metrics. However, this makes them less responsive than their counterparts, forcing them to retain a higher or lower power level longer than needed.

Both of these generic approaches do not take into account the interference they might produce by the frequent fluctuation of the T_x in a dense network consisting of multiple source nodes. Interference results in collisions and ultimately impacts the latency and PDR. The situation of collisions is magnified when the underlying routing protocols, e.g. Collection Tree Protocol (CTP) [36] are designed to periodically transmit control messages to keep the network connected. Medium Access Control (MAC) protocols such as ContikiMAC [76] are designed to retransmit the packets (control packets from routing protocol, data packets such as temperature or humidity from the application layer) until they receive the ACK from the receiving nodes, thus, increasing the energy consumption.

3.2.2 MAC Layer Protocol: ContikiMAC

The reason for not changing the MAC protocol in the experiments as we do with routing protocols is because ContikiMAC is known to outperform its popular predecessors such as B-MAC and variants of X-MAC (X-MAC-C, X-MAC-CP, and X-MAC-P) in terms of latency, retransmission, energy consumption and PDR [97]. The main features of this protocol are as follows [76]:

- It is an asynchronous sender initiated radio duty cycle protocol. Asynchronous meaning there is no common wake-up schedule established between a pair of nodes before the communication between them starts.
- Nodes running ContikiMAC continuously send the entire data frame until ACK is received from the recipient. This is unlike X-MAC where strobes are used by the sender and only after receiving corresponding strobe-ACK, the sender transmits the entire data frame.

- ContikiMAC does not have a fixed wake-up and sleep schedule. Instead, it has an adaptive scheme where the node performs two successive Clear Channel Assessments (CCA) to determine if there is an incoming data based on the RSSI. If the CCA finds the channel is clear, the nodes go to sleep. Else, the concerned node stays awake and executes a fast sleep optimization method. This method determines if the RSSI is due to noise or because of incoming data. If former is the case the node goes to sleep.

When the node incorporates TPC, the second feature of ContikiMAC can cause turbulence in the network. Imagine, a node continuously transmitting data for which it does not receive any ACK from the concerned receiver node (for e.g. due to channel busy on its side). A TPC based on the generic model will increase its T_x level causing contention in its vicinity. As neighboring nodes find that the channel is busy, they might defer their communication. This increases the latency. As the retransmission increases the channel utilization, the third feature of the ContikiMAC keeps the node in the wake-up mode for a slightly higher period at the expense of more energy consumption.

3.2.3 Routing Layer Protocol: ContikiMesh and CTP

For this experiment, we have chosen two routing protocols-ContikiMesh and CTP. The objective is to test the impact of real-time TPC on two variants of routing protocols - dynamic (CTP) and less dynamic (ContikiMesh) in terms finding and repairing broken links. ContikiMesh and CTP are known to generate least and highest amount of control packets respectively. Hence, both these protocols provide the opportunity to study the impact of TPC on the whole network under varying routing overheads. Furthermore, CTP is a widely used protocol for a static stationary network [98][8]. ContikiMesh, on the other hand, is used in a multi-hop network deployed in an open environment with little or no interference and traffic movement [14].

ContikiMesh is a lightweight protocol provided by the ContikiOS. Following are the important features of ContikiMesh [99]

- It uses two modules namely route-discovery and multi-hop to find the potential neighbours and multi-hops the data to specified receiver residing somewhere in the network.
- Once the route-discovery phase is completed there is no periodic transmission of control messages.
- It does not use any link quality estimation technique to cope with the dynamic environment.

CTP is a distance vector protocol that is capable of computing any-cast routes to a single or small group of sink in a network. The three most important features of CTP are as follows [36]

- Wireless links are unstable and exhibit bursty behaviour over the short time period. This suggests that the accuracy of link quality estimation can be high if it is agile. For this purpose, it uses information from the three layers namely physical, data link and network layers.
- It incorporates data-path validation scheme that reliably detects the path from the source to destination. The main task of this scheme is to avoid looping condition causing network congestion. For this, CTP uses probe packets to quickly detect the problem when the packets do not make progress towards the destination.
- Typical routing protocols transmit control messages at a fixed time interval. CTP, however, uses adaptive beaconing mechanism. When the topology is inconsistent, CTP transmits the control message faster and decreases it significantly when the network is stable.

Because of the third feature of CTP, TPC can cause more retransmission. Based on the default functioning of ContikiMesh and CTP in ContikiOS 2.7, we classify them as adaptive and non-adaptive protocols respectively. The adaptiveness of the protocols is tested in a scenario described in

Figure 3.1.

In

Figure 3.1 (a) and (b), node n_1 is the source node; n_2 and n_3 are the relay nodes. n_4 is the sink node. n_1 would either select n_2 or n_3 as a relay node to transmit its data to the sink. For explanation purpose, let us assume n_1 selects n_3 (shown by thick arrow line) as its relay to send the data. To simulate the link breakage between n_1 and n_3, we move n_3 out of the transmission range (dashed circle) of n_1 (shown by thin arrow line). The link breakage is detected at n_1 and it redirects its traffic to n_2 (shown by dashed arrow line). However, n_2 does not forward the data of n_1 to n_4. By placing n_3 back to its original position, the sink n_4 continues to receive the data of n_1 through n_3. However, in the case of CTP for the same network topology, the removal of node n_3 is detected at n_1 and it forwards the data to the sink via n_2 in few minutes by sending additional control messages.

Although ContikiMesh can save energy as there are fewer control messages, a link break can severely reduce the PDR of n_1 and also other leaf nodes (if present) that use n_1 as the relay node.

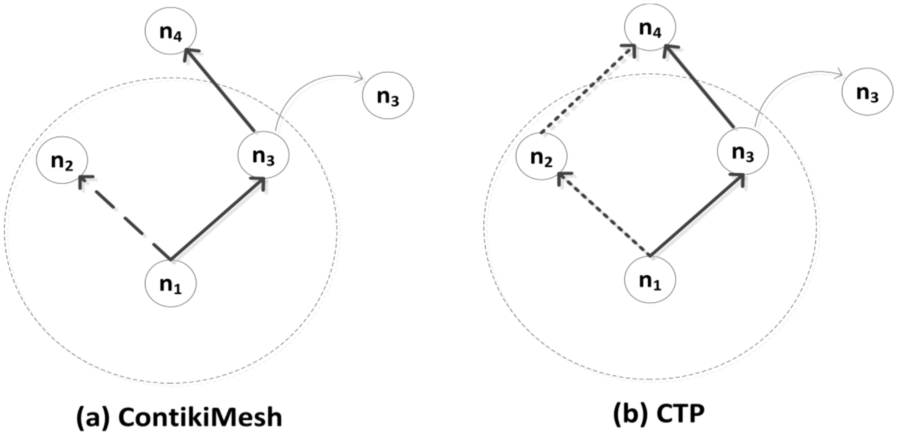


Figure 3.1 Routing of ContikiMesh and CTP

CTP, on the other hand, is more dynamic and has potential to detect routing problems. However, the disturbance in the network caused by TPC can force it to send higher control packets which consume more energy. In addition, under the flood of control messages, the probability of data packets not reaching its destination is higher.

3.3 Experimentation and Simulation Setup

In the following experiments, the goal was to study the impact of transmission power scaling on routing protocols. Therefore, we used a network emulation tool (Cooja Simulator [38]) to eliminate factors, other than power scaling.

3.3.1 Constraints and Requirements

While designing a TPC algorithm, importance should also be given to the application data rate. Consider a case where the TPC algorithm requires a considerable amount of historical link quality data to decide the future T_x to be employed. If the data rate is fast, the duration of time pervious T_x retained is small and the adaptation to variation is fast. The opposite is the case when the data rate is slow. The importance of performing TPC adaptation quickly is discussed in [100,101]. Having a high data rate does not necessarily translate into an increase in the throughput. This is even truer in a network with real-time TPC.

The way the nodes are deployed can also affect the performance of TPC. For example, in a dense network where the nodes are randomly deployed, a node cannot use a single global transmission power to reach its neighbours. Hence, all the nodes must perform prolonged initialization phase to determine the optimum T_x level on a per link basis. On the other hand, in a network of uniformly deployed nodes, nodes may employ less intensive initialization phase thereby saving energy. Furthermore, in a dense network with randomly distributed nodes the power level variations may be extreme. This can cause more contention than in a network with uniformly distributed nodes.

The placement of the sink also plays a crucial role. It is experimentally proven that merits of TPC are noticeable only in a network with convergence data traffic and not on a network where the traffic flow is aggregated [29]. The description of various network flows is in [102].

3.3.2 Network Topology

The experiments were carried out on a small sized homogenous spatially dispersed network as shown in Figure 3.2. Except for the sink node, all nodes in the network transmit one application data randomly in every 2 to 4 secs time interval. Fluctuation in T_x (from low to high and vice versa) resembles fast adaptation of T_x (6 every 1min). Because of high data rate and faster T_x adaptation, every experiment lasted for 15 minutes. There are in total 10 nodes, one sink, and nine independent sensor data source nodes. The longest path from data sources to the sink depends on the transmission power per node; for our experiments, it varies from 5 to 6 hops. The square grids in Figure 3.2 have edges 10m long. The average density of the network stands at around 65% and is calculated by using the equation 3.1

$$\sum_{i=1}^{n} \frac{D_i N}{100} \tag{3.1}$$

Here, D_i is the total number of neighbouring nodes for a given node with a specific transmission range and N is the total number of nodes in the network. The link between any pair of nodes is considered asymmetric. Node 10 is the sink node where the data is aggregated.

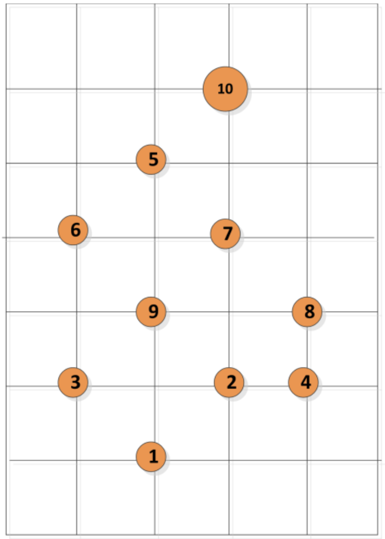


Figure 3.2 Topology of the network

To increase the confidence of our results and more reliably correlate transmission range with routing protocols performance, we placed the nodes at equal distances between each other. This ensures that the degradation of the network, if any, with respect to the reliability, latency, energy and packet loss is not due to the random placement of the nodes but the node transmission power scaling. The network size was kept limited for two reasons: (a) the routing paths are long enough to simulate the impact of transmission power scaling of a node to the remaining nodes of the path, and (b) the logged data were also stayed manageable making the traceability of network performance patterns more effective.

3.3.3 Transmission Power Scaling

The default transmission power of all the nodes is set to T_x=15. This corresponds to a transmission range of 22 meters, the output of -7dBm and energy consumption of 12.5mA as given by data sheet of CC2420 transceiver [71]. Transmission power upscaling boosts the power level to T_x=19. This translates to increase in the transmission range by 29 meters, the output of -5dBm and energy consumption of 13.9mA [71]. A lower power level 11 with a transmission range of 16 meters could have been used. However, this would place the nodes at the very edge of the radio range increasing the packet drop rate. This would erroneously affect the study.

Nodes 2, 7, 8 and 9 that have various interference coverages were chosen to perform the TPC. The interference area is divided into 1-hop region and 2-hop region. For example, in Figure 3.2, when the node 2 employs power level 15, the neighbouring nodes (1, 3, 4, 7, 8 and 9) that are within 22 meters region are said to be in 1-hop interference region and all the nodes beyond 22 meters range are said to be 2-hops interference region. Table 3.3 provides the details of the interference coverages of various nodes.

Table 3.3 Interference coverage of the nodes with two different transmission power levels

Power Levels	Descriptions	Node 2	Node 7	Node 8	Node 9
Default T_x=15	1-hop interference region (%)	67	78	45	89
	2-hops interference region (%)	33	22	54	11
Increased T_x=19	1-hop interference region (%)	78	89	67	89
	2-hops interference region (%)	22	11	33	11

Figure 3.3 illustrates a sample transmission power-scaling pattern. Data source nodes send sensor data to the sink randomly in 2 to 4 seconds time interval. That is, for every 10 seconds, these nodes generate 5-6 data packets. At the end of every 10 seconds period, the nodes 2, 7, 8 and 9 randomly decide to upscale (15 to 19), downscale (19 to 15) or maintain their transmission power. No specific TPC technique, e.g. Iterative method [103] or ATPC [68], is used. However, the focus is on the effect of transmission power scaling action; a study over the effect of random such actions is sufficient since, to a good extend, TPC based on the link quality metrics (Received Signal Strength Indicator and/or Link Quality Indicator) is mimicked.

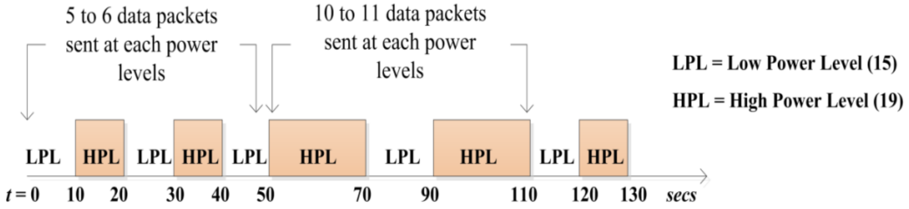

5 to 6 data packets sent at each power levels
10 to 11 data packets sent at each power levels
LPL = Low Power Level (15)
HPL = High Power Level (19)
LPL
HPL
LPL
HPL
LPL
HPL
LPL
HPL
LPL
HPL
t = 0
10
20
30
40
50
70
90
110
120
130
secs

Figure 3.3 Sample transmission power scaling pattern

Due to randomness, the 4 nodes that were scaling their transmission power had different patterns.

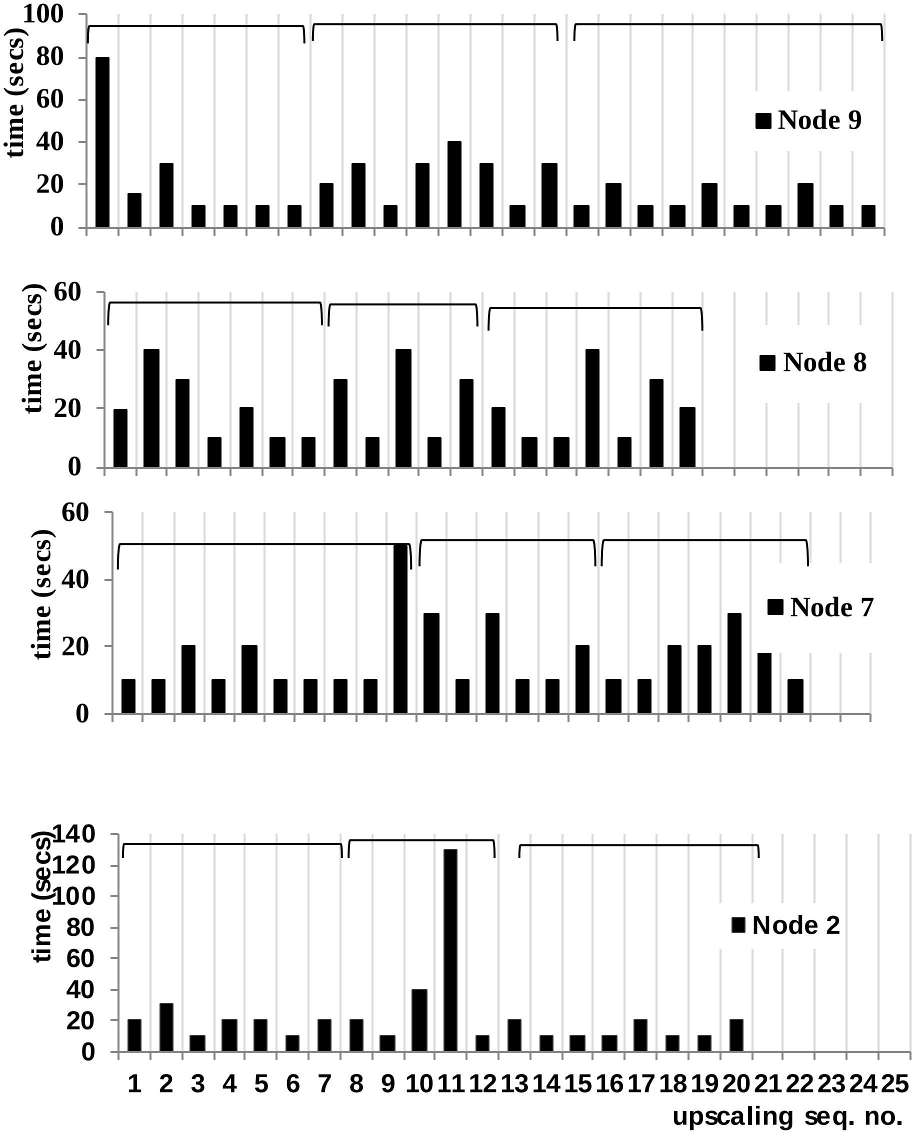


Figure 3.4 Duration of high transmission power after each upscaling event for ContikiMesh experiments

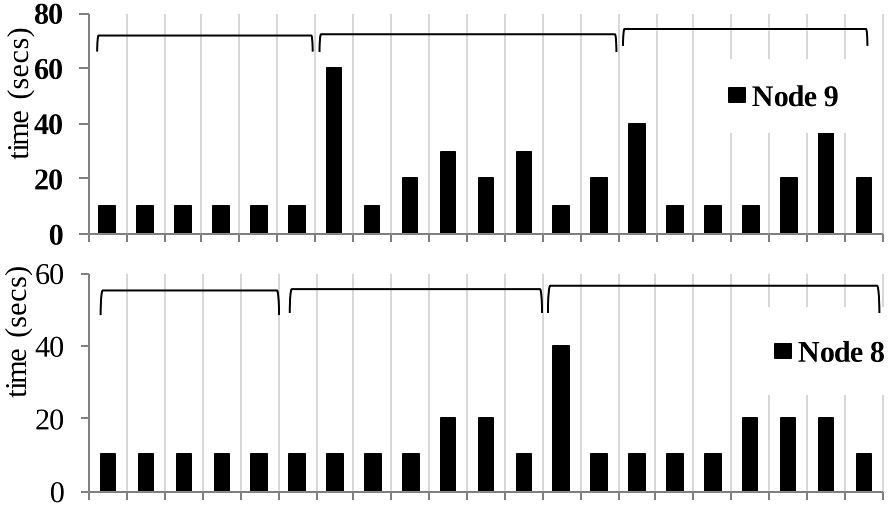
Figure 3.4 illustrates the duration a node maintained its high transmission power once it up scales for the ContikiMesh experiments. The upscaling events (x axis) are ordered based on the moment of occurrence. The duration of downscaling is not shown. On the one hand, node 9 seems to have the most upscaling events with a long high power start-up. On the other hand, Node 2 has few upscaling events of short duration except for one outlier 130secs long in the middle of that sequence.

Splitting the experimentation time of 15 minutes into three intervals of 5 minutes each, we generated Table 3.4 and Table 3.5. Tables illustrate the amount of upscaling events and the duration of high transmission power for each of the 4 nodes. These intervals are also marked in Figure 3.4.

Table 3.4 Upscaling events and duration of high transmission power per 5 min interval of ContikiMesh experiments

Descriptions	**Interval (mins)**	**Node 2**	**Node 7**	**Node 8**	**Node 9**
No. of times TPC was high (T_x=19)	0-5	8	10	7	7
	5-10	5	6	5	8
	10-15	8	7	7	10
Duration of high transmission power (mins)	0-5	2.33	1.99	2.33	2.66
	5-10	3.49	1.82	1.99	3.16
	10-15	1.82	2.16	2.33	2.16

The second set of experiments refers to Collection Tree routing protocol (CTP). The configuration is the same except for those parameters affected by randomness. Hence, the transmission power scaling pattern is shown in Figure 3.5 and Table 3.5.



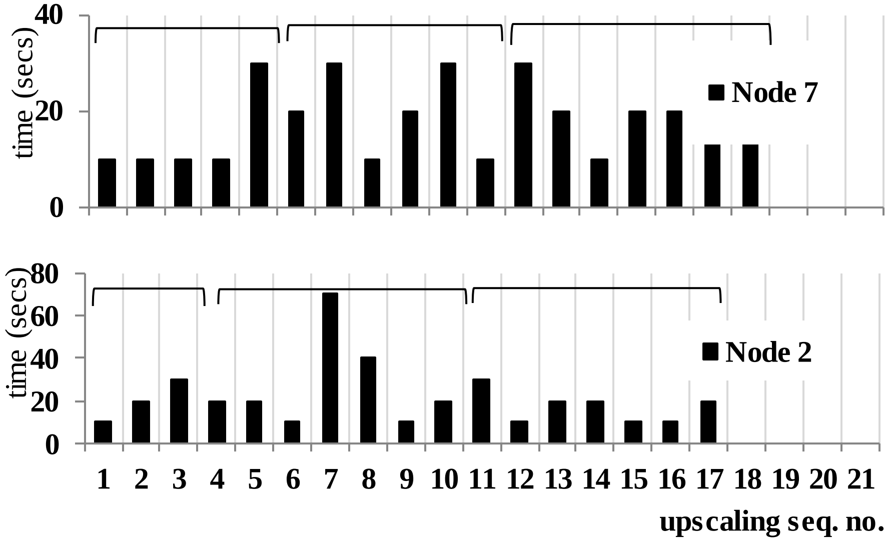


Figure 3.5 Duration of high transmission power after each upscaling event for CTP experiments

Table 3.5 Upscaling events and duration of high transmission power per 5 min interval of CTP experiments

Descriptions	Interval (mins)	Node 2	Node 7	Node 8	Node 9
No. of times TPC was high (T_x=19)	0-5	3	5	5	6
	5-10	7	6	7	8
	10-15	7	7	9	7
Duration of high transmission power (mins)	0-5	1	1.16	0.83	0.9
	5-10	3.16	1.99	1.49	3.33
	10-15	1.99	2	2.49	2.66

Figure 3.5 and Table 3.5 reveal a less dynamic transmission power scaling situation for CTP experiments compared to ContikiMesh ones. That is fewer power level changes and of shorter duration per level.

3.3.4 Emulation Features

Cooja was configured at 100% simulation speed. The radio messages of the network were propagated based on Unit Disk Graph Model (UDGM) and were captured using the inbuilt tool of Cooja. Based on UDGM, the strength of the signal fades with the distance between the source and destination nodes emulating link failures [49]. Below the chosen routing protocols, ContikiMesh and Collection Tree Protocol (CTP), the MAC layer was ContikiMAC.

All the nodes in the network have a start-up delay of 1000ms and emulate the TelosB sensor motes equipped with CC2420 transceiver [11]. To check the channel condition, all the nodes perform Clear Channel Assessment (CCA) at the MAC layer [76]. The battery depletion of the nodes has significant impact on the listening and transmission aspect of the nodes [32]. Therefore, battery depletion of the nodes is not simulated and nodes experience no power outage. Hence, we make sure that no disturbance is caused by factors other than transmission power scaling. Likewise, the receiver sensitivity of all the nodes is unchanged and is set to the default levels of -90dBm.

3.3.5 Benchmark and Assessment Metrics

Besides experimenting with the two routing protocols and different nodes (2, 7, 8, and 9) in power scaling mode, we have created a benchmark of no power scaling over the same network. That is two sets of 5 experiments per routing protocol, 10 experiments in total. The network topology and other configurations (except for random seeds and randomized parameters) are identical for both sets. The results of those two sets are separately analysed and finally compared together.

As explained above, transmission power scaling has a direct effect on the internal interference of the whole system. Indirectly, packet retransmissions, queue lengths, and energy consumption are influenced by the interference in the system. The question of this study is whether the adaptivity capabilities of routing protocols worsen or ease the situation. We have picked the following assessment metrics:

- *A number of packets* generated to assess how increased interference from the transmission power scaling node may cause collisions and trigger more retransmissions and or routing control packets.
- *End-to-end packet delivery ratio* (PDR) as a way to assess buffer overflows due to an excessive amount of interference that forces packets to stay in the queue for long and new ones to be dropped. This is the number of packets received at the sink over the total number of packets sent.
- *End-to-end latency* as an assessment of the queue lengths caused either by high interference and many retransmissions or by path rediscovery when paths are destroyed due to power scaling. We measure the average latency as the average time difference of a packet from its first transmission trial from the source node until the reception from the sink.

- *Radio duty cycle* to assess whether the energy savings from TPC outperform the energy costs introduced by increased signalling. It is the percentage of time the transceiver was on for the entire duration of the experiment.

3.4 Results and Discussions

The results provided are broadly classified into two main sub sections – impact of transmission power scaling on ContikiMesh, and on CTP. In both these sub sections, the analysis focuses on three main aspects: (a) impact of a single node's transmission power scaling to retransmissions of packets from nodes, (b) the impact to network level performance metrics, and (c) the benefits, if any, for the node in power scaling mode.

For clarity and brevity, the labels NO-TPC, 2-TPC, 7-TPC, 8-TPC, and 9-TPC in the following histograms denote the results of the experiments where node 2, 7, 8, and 9 performs TPC. The benchmark (no node in transmission power scaling mode) experiment is tagged as NO-TPC. The tag x-TPC maps to the experiment during which node x is in transmission power scaling mode.

3.4.1 Transmission Power Scaling over ContikiMesh

This section presents the traffic generated due to transmission power scaling in presence of ContikiMesh. The analysis of packets generated across the network includes the on-board sensors data, control packets generated by the routing protocols, and the ACK packets produced by the 802.15.4 protocol upon reception of the packets. Figure 3.6 illustrates the total and retransmitted number of packets generated in the network (a) as well as a breakdown of the total number of packets into 5mins segments.

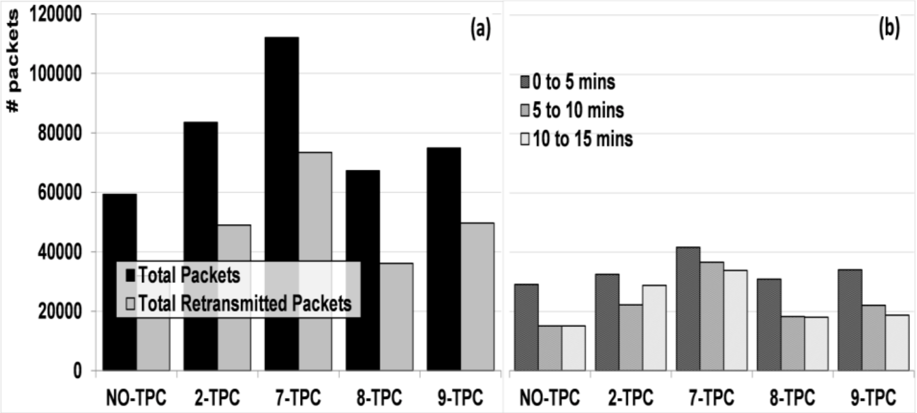


Figure 3.6 Packets transmitted during the 15 ContikiMesh experiments (NO-TPC & x-TPC). (a) Total number of packets and total number of retransmitted packets, (b) Breakdown of total number of packets transmitted into 5mins segments

The first segment of every experiment demonstrates a spike in the number of packets generated in the network compared to the remaining segments. The reason can be traced back to the route discovery phase of the ContikiMesh routing algorithm executed at the bootstrap of the network.

Compared to NO-TPC, the total number of packets generated in the network increases by 29%, 47.1%, 11.8% and 20.8% when nodes 2, 7, 8 and 9 perform TPC respectively. Similarly, transmission power scaling influences the amount of retransmitted packets as well. In fact, the total number of duplicates packet raises by 34.7%, 56.4%, 11.5% and 35.6% respectively; mostly higher than the increase in the total amount of packets.

When a specific node x up scales its power level (19), the neighboring nodes recognize the channel usage and defer the transmission of packets in their buffer. This results in fast buffer capacity consumption from own sensor data packets and relayed data packets from other nodes, too. All the deferred transmissions are retried as soon as the node downscales its power to the default level (15). That shortly increases collisions and nodes retransmit until ACK is received since in NO-TPC experiment, all the network features such as transmission power levels, data transfer rate, network topology etc. remain unchanged.

Amongst the experiments with transmission power scaling, the network of 8-TPC experiences the least amount of packets in the network; 7-TPC produces the highest followed by 2-TPC and 9-TPC. Duration of each power level, the frequency with which the levels interchange and the interference produced by the transmission power are responsible for the variation in a number of packets generated. Finally, retransmission trials increase with the betweenness centrality of the node in power scaling mode.

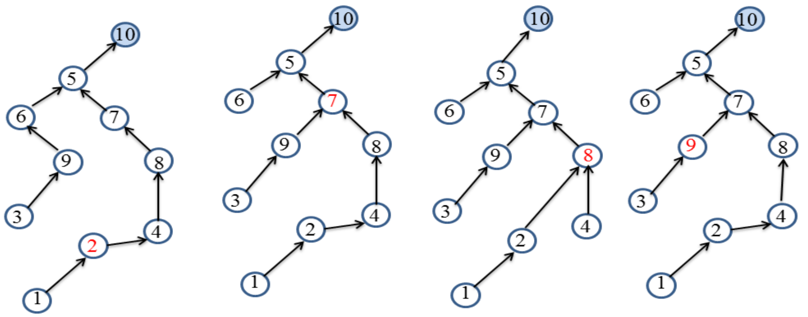


Figure 3.7 Routing paths built with ContikiMesh when nodes 2, 7, 8, 9 are in T_x scaling mode

Betweenness centrality refers to the amount of data traffic of other nodes relayed by a specific node in the network. For example, betweenness centrality of node 7 is the highest followed by node 8. Nodes 2 and 9 have the least betweenness centrality. Figure 3.7 and Table 3.6 illustrate the influence of betweenness centrality of nodes in transmission power scaling mode on a number of packets transmitted.

Table 3.6 Influence of betweenness centrality of nodes in transmission power scaling mode on packets generated

Descriptions	2-TPC	7-TPC	8-TPC	9-TPC
Relays the traffic of nodes	1	1, 2, 3, 4, 9, 8	1, 2, 4	1, 2, 3
Total packets sent compared to NO-TPC	+29%	+47.1%	+11.8%	+20.8%
Total control packets sent compared to NO-TPC	+34.7%	+56.4%	+11.5%	+35.6%

Although nodes 8 and 9 relay traffic from the same amount of nodes (3), the interference produced by node 8 in two different power levels, (15 and 19) is the least compared to all other nodes (see Table 3.3). Although, as shown in Table 3.4, the transmission power level of node 8 changes more frequently the duration per level is shorter than that of node 9. Therefore, neighborhood size increases with the transmission power upscaling directly affecting the number of collisions and retransmissions.

As of 7-TPC versus 9-TPC, the former experiences shorter duration per transmission power level and lower number of level interchanges than the latter. The interference produced during power level 19 is the same to 9-TPC (see

Table 3.3). However, in 7-TPC, node 7 relays double the traffic that node 9 relays in 9-TPC and therefore this result suggests that collisions and retransmissions increase with the betweenness centrality of a node in transmission power scaling mode even if the power upscaling is short and infrequent.

2-TPC comes to strengthen the conclusion that long duration at high transmission power levels creates an explosion of deferred packet (re-transmissions once the power is downscaled to low levels). As Figure 3.6 depicts, 2-TPC experiences 9% extra total number of packets even with lower interference region, duration per power level, the number of level interchanges and the amount of relayed traffic. 2-TPC generates more data packets in the last time segment (28785) compared to 9-TPC (19740), an increase of 34%. This is because, during the previous segment (5-10mins), node 2 remains in high power level (19) for 130 seconds. This forces nodes in the vicinity to defer for long before they are able to resend the packets and to cause an explosion of (re-)transmissions once the power level is downscaled.

In terms of the types of packets transmitted, we split Figure 3.6(b) into control signalling and data packets illustrated in Figure 3.8 respectively. Figure 3.8 confirm that the increased packet transmission in the beginning of every experiment is due to ContikiMesh route discovery signalling. Note that Figure 3.8 includes retransmissions.

9-TPC, as opposed to the other experiments, experiences more control packets than data packets even though data packets were generated at the same rate as any other experiment. Contrary to the other experiments, the node in the transmission power scaling mode (node 9) booted with high transmission power level and stayed in that level for 80secs. Therefore, the route discovery mechanism built the routing tables based on that power level. When later the node down scaled the power level, some of the routes had to be rediscovered. Moreover, this experiment has the most frequent power level interchanges among all. This deteriorated the situation as the route discovery was not allowed to converge and stabilize. The frequent power level changes also triggered multiple explosions of retransmissions from other nodes adding delay to the route discovery mechanisms or even loss of route discovery packets and unsuccessful route build-up. This resulted not only in a high number of control packets but also to undiscovered routes and, hence, fewer packet transmissions.

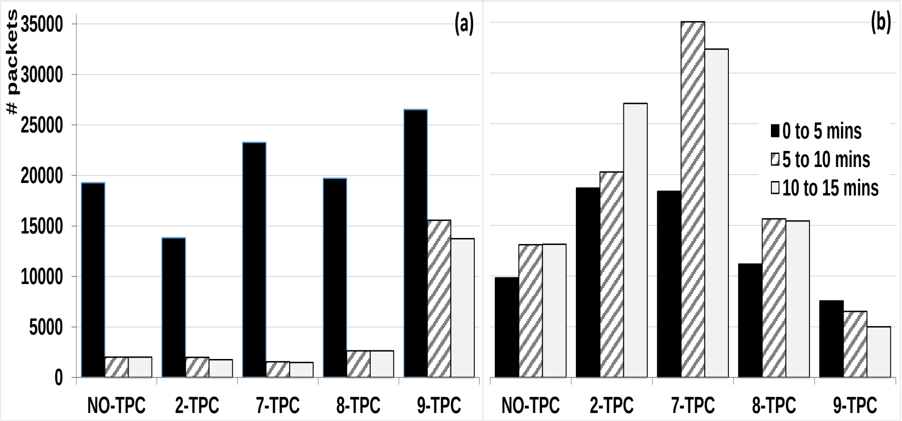


Figure 3.8 Total numbers of packets per packet type (control and data) per experiment and per time segment of 5 mins for ContikiMesh experiments. (a) Control packets, (b) Data packets

End-to-end network performance metrics: With regards to network performance metrics (end-to-end PDR & latency and duty cycle), the picture is mixed as shown in Figure 3.9. NO-TPC experiment demonstrates that constant T_x=15 for all the nodes is a more reliable option (higher PDR), more energy efficient (lower duty cycle) and average in end-to-end latency.

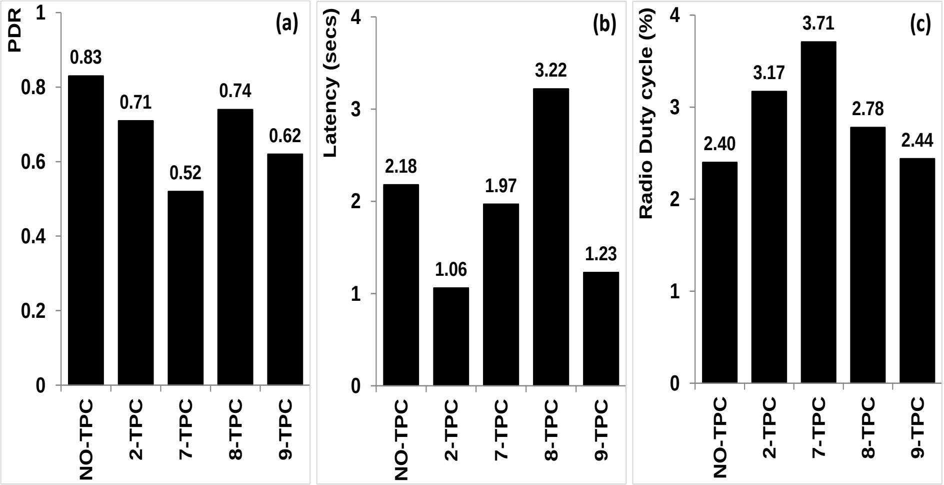


Figure 3.9 Average network performance metrics for ContikiMesh experiments. (a) End-to-end packet delivery ratio, (b) end-to-end latency, (c) radio duty cycle

As mentioned above, four important features of transmission power scaling vary the degree of the impact on the performance of the network. Of these four features, duration per transmission power level, neighbourhood size per level and interference region and the prominence betweenness centrality of the node in transmission power scaling mode have a major impact on the performance. This is noticeable in Figure 3.9. Although node 7 in 7-TPC experiment exhibits the smallest duration per level, it relays the traffic of most of the nodes. Node 9 in 9-TPC relays less traffic but the interference produced during high transmission power level is the highest among all the experiments. Nodes 2 in 2-TPC and 8 in 8-TPC relay less traffic and have smaller interference neighbourhood. Hence PDR is higher than the other two experiments.

As shown in Figure 3.9, in three out of four With-TPC scenarios, the latency is lesser compared to No-TPC. Small latency time is desirable property to have. However, recall that we calculate latency only for the number of packets that successfully arrive at the sink. As the environment is stable in No-TPC case, a number of packets that reach the sink is high. This is visible in the PDR value. Higher interference region due to high TPC causes a collision. Figure 3.9, combined with Table 3.3; indicate that the duty cycle increases with the interference neighbourhood. As collisions increase, packets are retransmitted and more energy is consumed. Since NO-TPC experiment experiences fewer collisions, the radio duty cycle of the network is 20.5% lower compared to all other experiments.

Selfish benefits of transmission power control: We have seen in the previous section how a single node that increases its transmission power to selfishly enhance its own one-hop link quality ends up adversely affecting the performance of the entire network. Here, we study what is the impact of this decision by a specific TPC node on its own overall performance. As shown in Figure 3.10, the average end-to-end PDR based on packets generated by nodes in transmission power scaling mode between all x-TPC experiments is 0.53. However, the average end-to-end PDR of the same nodes in NO-TPC experiment is 0.76, 30% higher. Similarly, as shown in Figure 3.10, the average end-to-end latency for the given nodes in x-TPC experiments is 2.05 secs and by 30% lower (1.27 secs) in NO-TPC experiment. The average duty cycle, Figure 3.10, is 7.52% and 2.8% in x-TPC and NO-TPC experiments respectively.

The experiments and results above give no evidence that reactive transmission power control provides benefits in the network or the node in transmission power scaling mode when ContikiMAC and ContikiMesh are used.

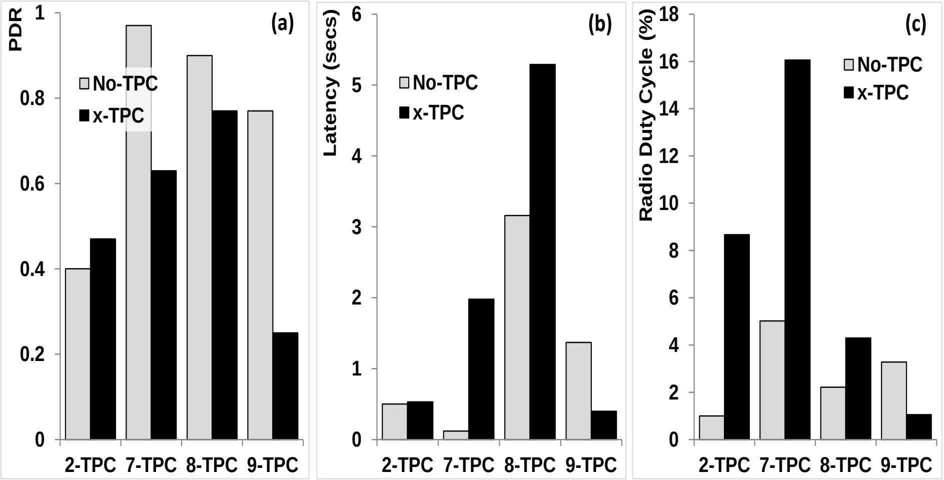


Figure 3.10 Average node performance metrics for ContikiMesh experiments. (a) End-to-end packet delivery ratio, (b) end-to-end latency, (c) radio duty cycle

3.4.2 Transmission Power Scaling over CTP

CTP is a dynamic routing protocol trying eagerly to re-establish more optimal routing paths taking into account the link quality between neighboring nodes. Figure 3.11 provides the routing paths in the network as established during the experiments.

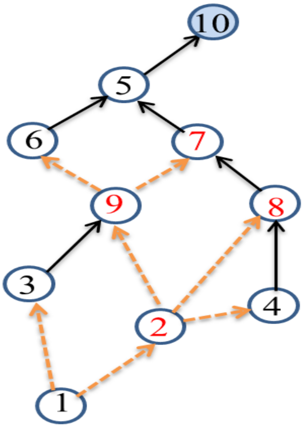


Figure 3.11 CTP routing path of the network when nodes (2, 7, 8 or 9) up/down-scale their transmission power

The dashed lines show the possible different paths a packet from a node can take depending on the quality of each link. The solid line shows the route from the respective nodes even in the presence of nodes 2, 7, 8 and 9. This comes with the cost of extra signaling, control packets to maintain a good path. Transmission power control generates two opposing forces to the performance of a network. From one hand, link quality may be improved and on the other hand, more interference due to extra signaling might deteriorate the performance. We follow the same analysis strategy as with ContikiMesh.

Traffic generated due to transmission power scaling: The total number of packets generated by CTP experiments, Figure 3.12(a), shows a clear difference from the ContikiMesh experiments. The total and retransmitted number of packets is higher. Especially, the ratio retransmitted a total number of packets is almost twice as big as in the ContikiMesh experiments. As pointed out from Figure 3.5 and Table 3.5, the transmission power scaling is less dynamic in CTP experiments, yet the number of generated packets higher. Moreover, from Figure 3.12, it is also clear that the route discovery is not taking place only at the bootstrap time since the first time 5 mins segment experiences the fewest packets compared to the other two segments.

Based on Table 3.7, transmission power scaling seems to have a positive effect on the number of packets for some experiments. There is a decrease in the total number of packets sent during 2-TPC, 8-TPC, and 9-TPC experiments. 7-TPC experiences the least power level changes, yet, the highest increase in a total number of packets. As shown in Table 3.6 and Table 3.7, in the 7-TPC experiment, node 7 relays the traffic from the majority of nodes and the interference produced by it is the second highest compared to all other nodes in transmission power scaling mode. Although node 9 in 9-TPC has fewer relay nodes, the interference produced by it during TPC is the highest, resulting in higher packet transmission compared to NO-TPC. The opposite is the case with 8-TPC, less interference fewer retransmissions. As with ContikiMesh experiments, betweenness centrality and interference neighborhood size of the node in transmission power scaling mode seem to play an important role in the number of packets generated. However, the results of both sets of experiments indicate that betweenness centrality has a stronger effect than interference.

Unlike ContikiMesh with its fixed beaconing procedure, CTP detects prevailing conditions in the radio propagation medium and employs adaptive beaconing of control messages. CTP mitigates the problem of self-interference by limiting transmission rate; i.e. the expected time of the packet to be sent is p, then CTP delays the packet transmission in the range of 1.5p to 2.5p. This adaptive behavior allows transmission power scaling generate benefits for the network in low self-interference situations i.e. 2-TPC and 8-TPC.

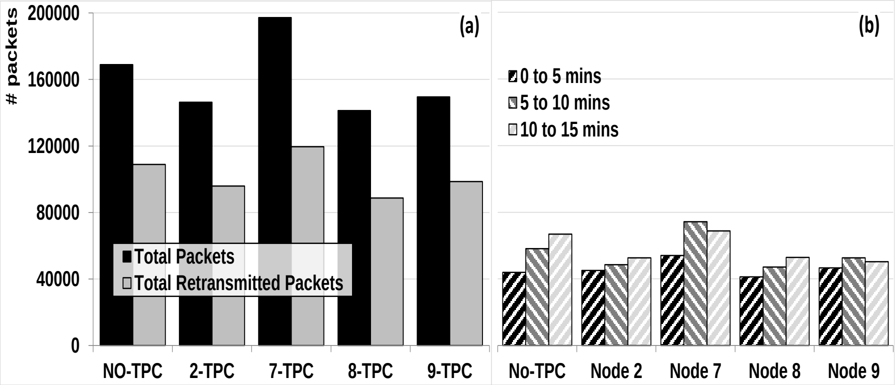


Figure 3.12 Packets transmitted during the 5 CTP experiments (NO-TPC & x-TPC). (a) Total number of packets and total number of retransmitted packets, (b) Breakdown of total number of packets transmitted into 5mins segments

Table 3.7 Influence of betweenness centrality of nodes in transmission power scaling mode on packets generated

Descriptions	2-TPC	7-TPC	8-TPC	9-TPC
Relays the traffic of nodes	1	1, 2, 3, 4, 9, 8	1, 2, 4	1, 2, 3
Total packets sent compared to NO-TPC	-13.39%	+34.73%	-28.31%	+5.78%
Total control packets sent compared to NO-TPC	-16.08%	+54.15%	-32.20%	-3.41%

As depicted in Figure 3.12(b), there is a hike in packets in the first five minutes due to route discovery, which largely constitutes control packets. Figure 3.13(a) illustrates the number of control packets transmitted per 5-min segments. With the exception of 2-TPC, the first 5min segments indeed experience more control packets compared to the following two segments. The situation is reversed with the data packets, as shown in Figure 3.13(b). As shown in Figure 3.5 and Table 3.5, node 2 in 2-TPC (5-10min segment) experiences a prolonged high transmission power level. This creates reliable links to nodes further away from node 2. It reduces the ETX to nodes at a longer distance and, hence, rewiring and route re-discovery are triggered so that shorter paths are built. This process generates more control packets but shortens the paths and reduces retransmissions; thus, fewer data packets.

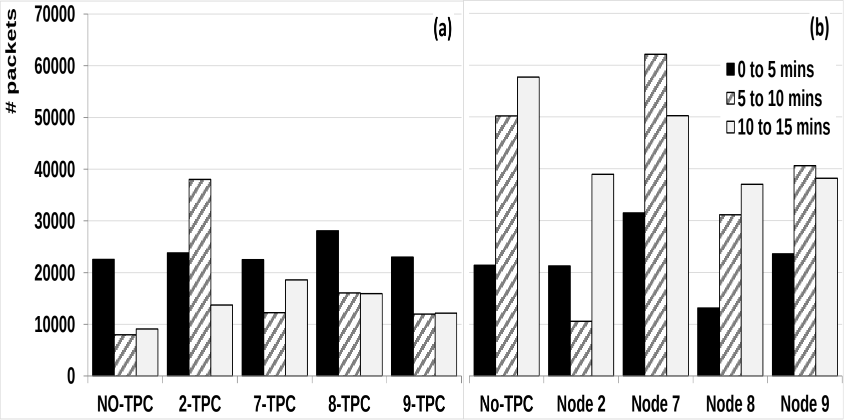


Figure 3.13 Total number of packets per packet type (control and data) per experiment and per time segment of 5 mins for CTP experiments. (a) Control packets, (b) Data packets

End-to-end network performance metrics: The adaptive beaconing of CTP cooperates with transmission power scaling and yields improvements with regards to network performance compared NO-TPC situation. Figure 3.14(a), Figure 3.14(b) and Figure 3.14(c) illustrate the average end-to-end packet delivery ratio, average end-to-end latency, and the average radio duty cycle, respectively. PDR has improved (+16.6%) in x-TPC experiments, and the latency is reduced (-18.6%) and the radio duty cycle is mostly reduced (-8%, except for 7-TPC).

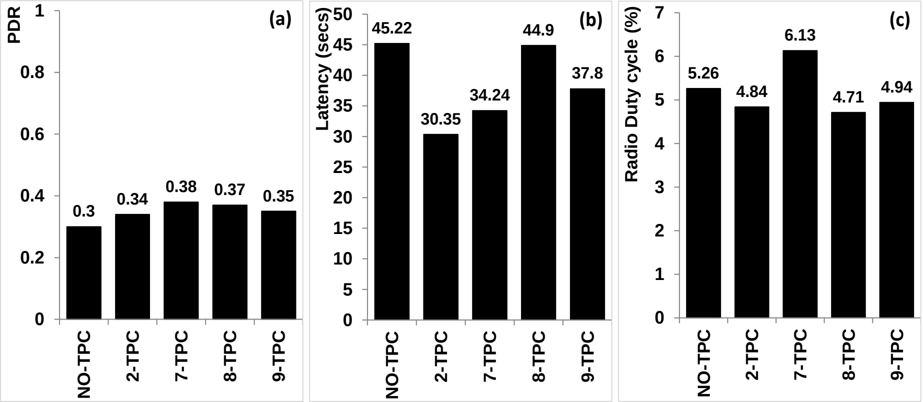


Figure 3.14 Average network performance metrics for CTP experiments. (a) End-to-end packet delivery ratio, (b) end-to-end latency, (c) radio duty cycle

Selfish benefits of transmission power control. While transmission power scaling has a positive effect on the overall performance of the network, Figure 3.15 illustrates the marginal benefits (PDR, latency and duty cycle) it has on the node in transmission power scaling mode. It is important here to note that transmission power scaling was a random process and did not simulate careful decisions to upscale or downscale transmission power based on some measurable metric.

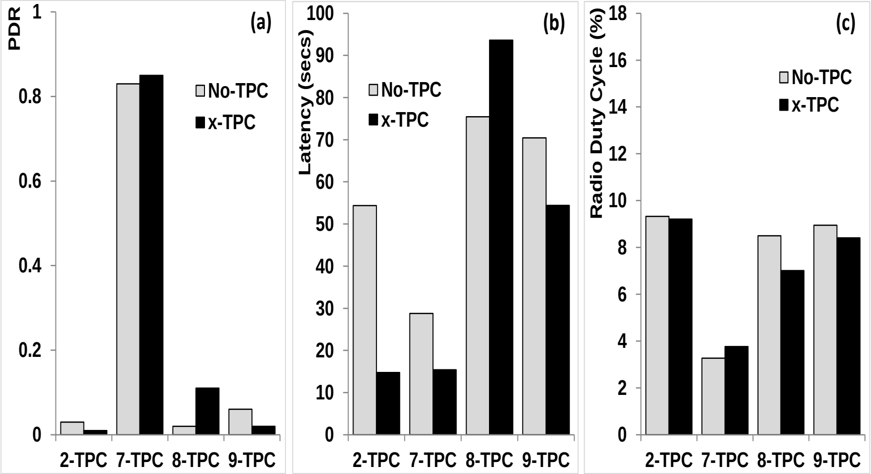


Figure 3.15 Average node performance metrics for CTP experiments. (a) End-to-end packet delivery ratio, (b) end-to-end latency, (c) radio duty cycle

3.4.3 Transmission Power impact on ContikiMesh and CTP

Between the two sets of experiments (ContikiMesh versus CTP), transmission power scaling seems more compatible with the latter. The periodicity of the control packets in CTP prevents the protocol from overreacting to transmission power level and link quality changes. In fact, CTP was designed to precisely handle these situations.

ContikiMesh triggers route re-discovery once it is too late and the route to a destination is lost. Down scaling the transmission, power may make routes built during high power level undiscoverable. Hence, the nodes wait until either the routes are rebuilt or power levels re-up scaled. These waiting period forces nodes, especially those with high betweenness centrality, generate a burst of packets once the route is available again. This burst increases temporarily the internal interference that results to CCA attempts and retransmissions. The phenomenon is worse when the scaling is frequent because even route discovery processes might be split between different power levels. That results to partial invalid routes. Therefore, transmission power scaling under ContikiMesh generates no benefit for either the network or the node in scaling mode.

On the other hand, the results on CTP demonstrate slight and under certain conditions improvements in the total number of packets, PDR, latency and duty cycling for both network and node performance. The experiments suggest that the transmission power should (a) be lower as the betweenness centrality increases, (b) change as rarely as possible, and (c) scale up and down if the routing protocol is dynamic (not on-demand route discovery) and periodically update the routes.

For completeness, Table 3.8 provides the standard deviation of metrics for all the experiments. The fluctuation in metrics across all the experiments indicates that the impact of TPC is prominent in a static multi-hop network. The design of TPC is strongly influenced by the traffic flow, data rate, the nature of node deployment, the routing and MAC protocols. Failure to study these influences can lead to a TPC that can function only in a specific scenario making its usage limited. Lastly, we do not recommend reactive TPC algorithm to be designed based on the RSSI or LQI. The reason for this is that these parameters are sensitive to environmental changes and therefore can trigger TPC frequently causing the network performance to drop. Moreover, RSSI and/or LQI vary from location to location. Hence, TPC based on these metrics are not location agnostic requiring a considerable amount of fine-tuning of the threshold level before the network becomes operational. In addition, a reactive TPC algorithm can often keep the transmission power at a high level until the next batch of link quality metrics is obtained. From our experiments, we found that higher the duration of high transmission power, higher is the chance of collision.

Table 3.8 Standard deviation of metrics for two routing protocols

Metrics	NO-TPC		x-TPC							
	Contiki Mesh	CTP	ContikiMesh				CTP			
			2-TPC	7-TPC	8-TPC	9-TPC	2-TPC	7-TPC	8-TPC	9-TPC
PDR	0.3	0.41	0.32	0.28	0.28	0.43	0.47	0.41	0.43	0.46
Latency (secs)	1.95	21.58	1.34	2.07	2.73	0.91	27.80	29.11	38.08	27.39
Radio Duty Cycle (%)	1.16	3.19	2.24	4.40	1.39	1.05	2.69	1.72	2.6	2.70

3.5 Conclusions and Future Work

In this chapter, we studied the impact of frequent TPC on a static multi-hop network consisting of simultaneously transmitting nodes. The data traffic in the network flows on a routing tree aggregated to a single sink. We found that when a node scales its transmission power, the impact to the entire network performance in terms of PDR, latency and energy consumption is significant. This is true in both ContikiMesh and CTP over ContikiMAC. In the former case, the PDR and latency of the network drop by 21% and 14%, respectively, and the energy consumption increases by 20% compared to the NO-TPC scenario. Fall in the latency here is because a lesser number of packets arrives at the destination. Compared to No-TPC scenario, there is an increase in a number of total packets and retransmitted packets sent by 29% and 38% respectively. With CTP and ContikiMAC we find that the PDR of the network increases by 16% and there is a decrease in the energy spent by 1.9% and the latency by 43% compared to the No-TPC scenario. There is also a drop in total and retransmitted packets by 6.13% and 7.5% respectively.

Although between two routing protocols, CTP is immune to the turbulence caused by TPC in the network; overall PDR achieved for energy consumed is not encouraging. One way to reduce additional routing messages is to use a variant of AODV routing protocol known as Gossiping based AODV [104] or utilize a routing protocol based on congestion metric [105]. Performing the same experiments and evaluation methodology for AODV and RPL is planned for future work. Another way to overcome this drawback is to use TPC in conjunction with a scheduler that carefully allocates different timeslots and/or channels for every pair of nodes in the network. Although this technique requires complex scheduling techniques [106] that adhere to strict deadlines, it may be the only way to reduce collision and retransmissions. We carried out our experiments in a simulator with an ideal circular radio model on a homogenous network. In reality, the radio ranges are highly irregular and the multi-path effect and a random node deployment can produce a result inferior to what we already have. Conducting the same experiment with real hardware and deployment such as office space is also planned for future.

Our conclusion is that TPC inadvertently influences routing and MAC layers. In addition, data rate, traffic flow, and node deployment also have a significant impact on the working of TPC. Failure to study these factors can lead to a design of TPC that can do more harm than benefit the network. An isolated design of TPC may lead to non-generic TPC that works only under specific condition and location. This requires time consuming fine-tuning of TPC before the network becomes operational.

Chapter 4 Reliable Low-Power Wireless Networks over Unstable Transmission Power

To reduce the energy cost incurred during transmission, sensor nodes in IoT (Internet-of-Things) use lower transmission power ($LowT_x$). However, quality of the link achieved by a pair of nodes through $LowT_x$ increases the chance of an unstable link. In addition, it aggravates hidden-node terminal and energy-hole problems. Although nodes employing higher transmission power ($HighT_x$) reduce the issues encountered in utilizing $LowT_x$, it may consume more energy. Therefore, the main research objective of this chapter is to address the following research objective mentioned in chapter 1

(R03) *Investigate if there is a trade-off in terms of reliability and energy consumption in adoption of $LowT_x$ and $HighT_x$ by the sensor nodes.*

In the previous chapter, we investigated the impact of one node performing TPC on the overall network. In this chapter, we study the overall performance of the network of various densities in terms of Packet Delivery Ratio (PDR), Radio Duty Cycle and collision in two different scenarios – network wide fixed and random application data rate transmitted at varying transmission power (T_x) levels.

Our experiments show that compared to $HighT_x$, at $LowT_x$, duty cycle and collision in two different network densities is higher. Similarly, the PDR achieved at $HighT_x$ is higher than that of $LowT_x$.

We perform experiments in 802.15.4 network of various size and deployment. The experiments were carried out in COOJA which emulates TelosB nodes based on a CC2420 transceiver [38]. We study the performance of the network by means of three metrics–Packet Delivery ratio, radio duty cycle and collision rate in the presence of network wide fixed and random application data rate transmitted at varying transmission power (T_x) levels. Our study shows that $LowT_x$ fares poorly compared to $HighT_x$ in terms of PDR, energy consumed (radio duty cycle) and collision rate.

The remainder of the chapter is organized as follows. Overview of link quality at low transmission power is presented in Section 4.1. In Section 4.2, we present the related work. General demerits of $LowT_x$ are provided in Section 4.3. Section 4.4 provides the information about experimental setup and the simulation parameters used. Discussion on the results is provided in Section 4.5. Finally, we conclude the chapter in Section 4.6.

4.1 Link Quality at Low Transmission Power

IoT is revolutionizing many sectors such as logistics and transportation, manufacturing plants and agriculture by increasing their operational efficiency, reducing the energy usage and better management of resources. As IoT provides immense benefits, it is becoming a competitive advantage and the business houses have to consider it as a part of their strategic plan. As a result, the demand for IoT is rising.

Irrespective of the IoT deployment scenarios, communication reliability of the deployed IoT is often affected by ever changing propagation medium. The uncertainty in propagation medium is mainly caused by the following factors:

- Natural obstacles such as office walls, human and vehicular movements, dense vegetation generally deteriorate the signal strength [20].
- Environmental conditions such as humidity and temperature also influence radio waves [21][22][23].
- Orientations of the antenna, remaining battery life of the nodes are also known to alter the radio wave [27][32].
- Interferences from Wi-Fi routers, microwave ovens, and other co-located sensor network can drastically impede the link reliability [16].

These negative factors have an amplifying effect on weak communication link achieved by $LowT_x$ than $HighT_x$. Consider a source (S_n) and receiver (R_n) TelosB sensor nodes separated by a distance of 10m as shown in Figure 4.1.

A transmission range (T_r) achieved by a certain T_x is known to be $LowT_x$ if S_n can successfully transmit the packets to R_n with lower energy expense. For instance, the power level parameter T_x =11 (shown by dashed line) of TelosB node can be classified as $LowT_x$ because it results in T_r of 16m and consume 11.2mA (refer Table 4.1) [71]. Although T_x at 15,..., 31 that result in T_r of 22m, 48m respectively puts R_n within the transmission range of S_n, it consumes 12.5mA, 17.4mA energy respectively and therefore it is classified as $HighT_x$.

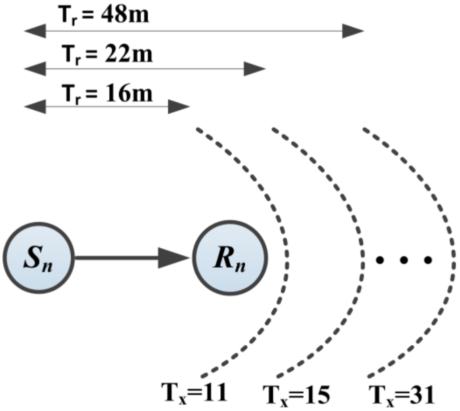


Figure 4.1 Transmission range of TelosB sensor nodes

When the source node employs $LowT_x$, the receiving node may not receive the data due to above-mentioned factors or due to internal interference and hence no ACK is sent back. This causes the source node to retransmit the data repeatedly until ACK is received or forces the routing protocol to change the route at the expense of transmitting more control messages. The new route chosen may not be efficient path to relay the data to its destination. In both cases, more energy is consumed. Therefore, minimizing transmission power and simultaneously strengthening the communication link are conflicting goals [34].

To enhance the link reliability/QoS between the pair of nodes, boosting the T_x seems to be a natural choice. However, this may introduce interference. In addition, generally, the transceiver of several sensor nodes such as TelosB[11], MicaZ[12], etc. utilized in IoT network consumes more energy than other units such as ROM, sensors, CPU of the nodes [71]. $HighT_x$ does reduce the number of hops data takes before it reaches its destination and hence latency is reduced.

IoT consists of network-embedded devices that operate on a limited battery power. Remote deployment of IoT network makes it difficult for a field technician to replace the battery sources. Hence, prolonging the battery life of the sensor nodes is crucial. However, extending the battery life without delivering the data with decent reliability fails the purpose of the sensor network.

In realistic IoT networks, there may be hundreds of nodes simultaneously communicating among each other within relevant multi-hop cluster networks. Thus, it is crucial to study the collision rate, energy consumption, and reliability of the network at different T_x levels.

The findings of this research could be used as a key decision point in designing a robust Transmission Power Control (TPC) algorithm that can adjust the T_x according to a varying condition in the propagation medium. The majority of the TPC algorithms such as P-TPC, ODTPC, and BS-TPC increase T_x step-wise as and when the link quality of the propagation medium varies. [55][57][59]. However, increasing the T_x to the next level may not necessarily enhance the link quality. The output of the experiments conducted in this chapter could aid the TPC designers to select much higher T_x and thus avoid recipient nodes from piggybacking control packets.

4.2 Related Work

The effects of various transmission power levels on energy consumption are investigated by using a linear programming framework and a well-known continuous transmission power model (HCB model) in [107]. The authors through mathematical model show that in the high-density network, utilization of maximum transmission power level is a better option than lower transmission power and it does reduce the network lifetime. They use MicaZ mote (with MPR240 transceiver) energy model[12] and do not consider energy consumed due to routing overhead. Our experiment differs from them as we use TelosB sensor motes with the CC2420 transceiver and have multiple sensor nodes transmitting data simultaneously and we use COOJA emulation environment [38].

The performance of $HighT_x$ and $LowT_x$ in mobile ad-hoc network is evaluated in [108]. The experiments conducted in a network comprising MicaZ motes show that the energy consumption and PDR achieved at $LowT_x$ is only marginally better that its counterpart.

The trade-off between $LowT_x$ and $HighT_x$ in terms of throughput and energy consumption for indoor 802.11a network shows that $LowT_x$ along with virtual carrier sensing enabled at MAC level degrades the throughput of the network [34]. However, our experiments are carried out in 802.15.4 network.

TPC algorithms such as ATPC, Distributed TPC, and Probe-based TPC employ lengthy initialization phase to find appropriate T_x level to transmit data under non-varying propagation medium[68][109][41]. When there is a variation in the medium, TPC algorithms such as P-TPC, ODTPC, AODTPC increases the T_x to next level [55][57][58]. However, increasing the T_x to the next level may not necessarily enhance the link quality. When T_x adopting to the next level does not enhance the link quality, then pair of nodes in communication involve in piggybacking control packets that consume more energy. Worse, two nodes in communication may take a long time to converge to a specific T_x. In addition, it is shown that constant fluctuation of T_x in a network with multiple transmitting nodes reduces the PDR and increases the chances of collision [110]. The output of the experiments conducted in this paper could aid the TPC designers to select much higher T_x and thus avoid lengthy initialization phase.

4.3 Drawbacks of Low Transmission Power

4.3.1 Latency is higher

Naturally, by adopting $LowT_x$, data takes several hops before it reaches its destination and hence there is an increase in latency. As IoT network often have lossy links due to environmental factors, the communication delay is highly uncertain and can alleviate latency [111]. For surveillance and fire detection systems, the event of interest has to be communicated to the concerned authority with very low latency [112][113].

4.3.2 Energy-Hole problem

Consider a network of seven nodes as shown in Figure 4.2. When all the nodes choose $LowT_x$ (shown by double arrow lines), the data has to take many hops and hence the traffic D_i increases substantially near the sink as the nodes relay more data [28]. Node 6 near the sink relays five times (excluding its own data) more traffic than all other remaining nodes. This causes the nodes near the sink to expend more energy than others resulting in energy hole [114].

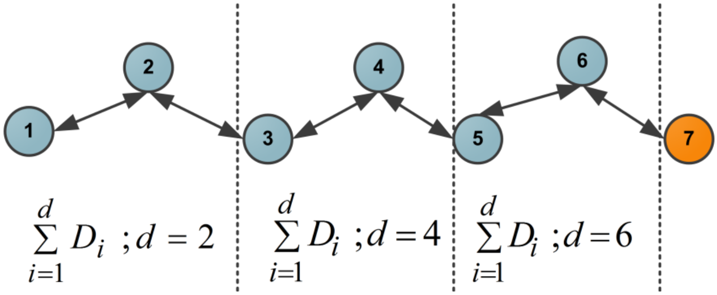

2
4
6
1
3
5
7
$\sum_{i=1}^{d} D_i \;; d = 2$
$\sum_{i=1}^{d} D_i \;; d = 4$
$\sum_{i=1}^{d} D_i \;; d = 6$

Figure 4.2 Traffic pattern at low transmission power level

4.3.3 Hidden-Node problem

The chances of a collision in the network with the node using $LowT_x$ is more compared to the node employing $HighT_x$. This is illustrated in Figure 4.3. Consider a scenario where node 1 is transmitting data to node 2 with transmission range $T_r = T_1$ and similarly node 3 is transmitting to node 4 with $T_r = T_2$. As node 1 and 3 are not within their respective T_r, they can communicate with node 2 and 4 respectively.

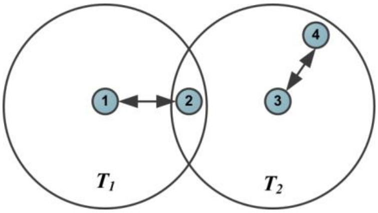


Figure 4.3: Illustration of hidden-node problem at lower transmission power

However, there is collision happening at node 2. Due to the collision, node 2 is not able to send ACK packet to node 1. As ACK is not sent to node 1 by node 2, node 1 retransmits the data packet until it receives ACK from node 2. This is often the case when a specific node varies its transmission power dynamically (refer Chapter 3). Depending on the functioning of MAC protocol, the amount of retransmission varies. Higher the retransmission, more energy is expended and there is huge packet loss [115]. One of the simple solutions is to increase the T_r of either of the nodes such that they fall within their range. For instance, when node 1 increases its T_r node 3 detects channel busy condition and will delay its transmission to node 4. The amount of packet loss and retransmission because of $LowT_x$ is provided in section 5.

4.3.4 Breakage of Low-Power Link is higher

It is a proven fact that low-power links are more susceptible to noise and multi-path distortion [16]. Temporal links are prone to link failures causing the routing protocols to generate more control packets to repair/find more stable links [92][93]. This advertently causes a collision and hence the energy depletion is faster than expected. Consider a real-time IoT network used for collecting vital biological signs of patients in hospital [8]. Any delay in transmitting this information to the concerned technician may result in loss of life. In these situations, using low-power links may not be appropriate.

4.4 Experimental Setup and Simulation Parameters

In this section, we provide a detailed explanation of our experimental setup and the simulation parameters used to test the performance of fixed $LowT_x$ and $HighT_x$.

4.4.1 General Characteristics

Following simulation parameters are kept constant for all the experiments.

- We validate the results for the static network. Once the respective experiments are started the position of nodes are unchanged.
- Cooja simulator emulating the functionality of TelosB equipped with the CC2420 transceiver is used. TelosB uses Contiki 2.4 OS.
- To emulate the link failure due to fading of the signal strength as the distance between the nodes increases, we use Unit Dish Graph Model (UDGM)[49].
- The duration of experiments is 20 minutes and is run with the 100% simulation speed.

4.4.2 Network Deployment

We use two network deployments in our experiments to validate our research objectives. These deployments are derived from real-world implementation. The transmitter of TelosB, CC2420 can be configured to use mainly eight different power levels. The output dBm, energy consumed and the distance corresponding to each level is provided in Table 4.1. These values were obtained from Cooja simulator.

From Table 4.1, we can infer that for every increase in the transmission power level, output power varies in the range of 1.3 to 1.4 mA. Similarly, the transmission range increases approximately by 6 to 7m and interference for T_x respective doubles.

Table 4.1 Power consumption and communication range of CC2420 transceiver

Power Levels Parameters	**3**	**7**	**11**	**15**	**19**	**23**	**27**	**31**
Transmission Distance (m)	3	10	16	22	29	34	42	48
Interference Distance (m)	6	20	31	42	58	68	83	96
Output Power (mA)	8.5	9.9	11.2	12.5	13.9	15.2	16.5	17.4

Linear deployment of nodes

Figure 4.4 shows the linear deployment of 15 nodes distributed across 140X10 meters. The distance between the nodes (separated by dashed and solid lines) is 20 and 10 meters respectively. Node 15 represents the sink node and therefore the data traffic is multi-hopped and aggregated at one point. This type of deployment can be found for instance in monitoring the structural health of Golden Gate Bridge, US [13].

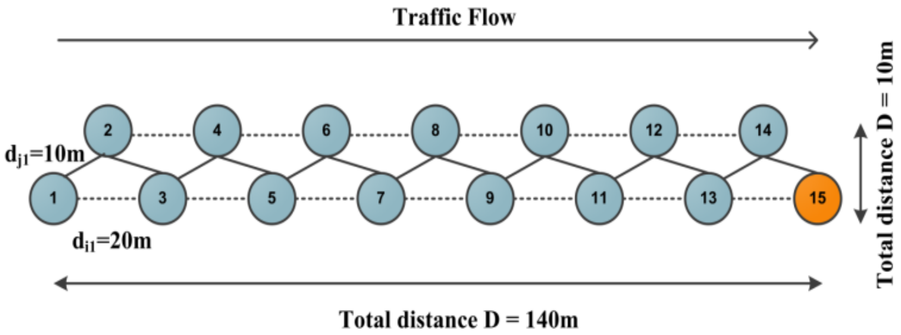
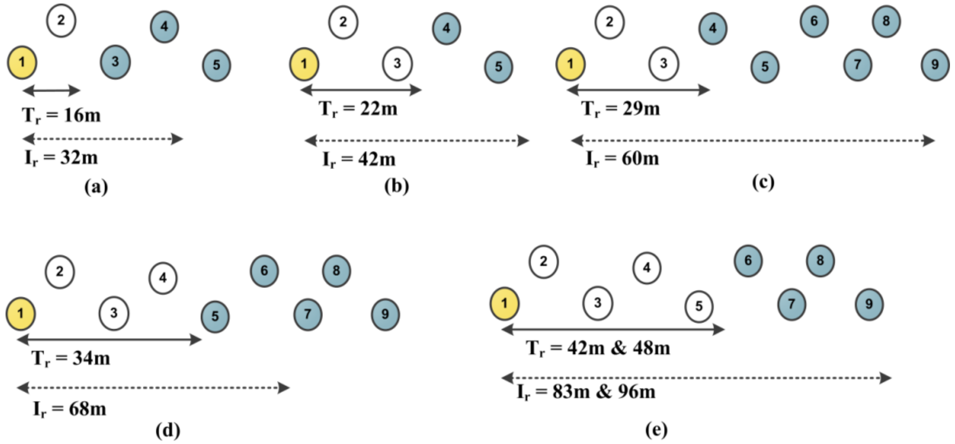


Figure 4.4 Linear deployment of nodes



(a) (b) (c) (d) (e)

Figure 4.5 Transmission and interference range obtained for various power levels

Figure 4.5 provides the information about the number of nodes in the transmission range (T_r) and interference range (I_r) at different power levels for node 1. As we can see from Figure 4.5 that the interference range (dashed line) is almost double than the transmission range (solid line). From Figure 4.5, we can get the general perception of how the transmission and interference range could be for other nodes. The transmission range directly influences the routing table of specific source node transmitting data to the final destination. Figure 4.5 (a) and 4.5 (b) shows the range (transmission and interference) of T_x levels 11 and 15. Figure 4.5 (c and d) shows the range corresponding to the T_x levels 19 and 23. Similarly, Figure 4.5 (e) represents the T_x levels for 27 and 31. The

T_x level 3 and 7 (refer Table 4.2) are not used as their respective transmission range is less than the 10m.

Hybrid deployment of nodes

Figure 4.6 depicts 30 nodes network distributed in 70X90 m with one sink node (30) placed at the center. This network is hybrid because the data traffic is both point-to-point for nodes nearer to sink and multi-hop for nodes away from the sink. This deployment is similar to the wireless sensor network (WSN) deployed to monitor the ocean [15].

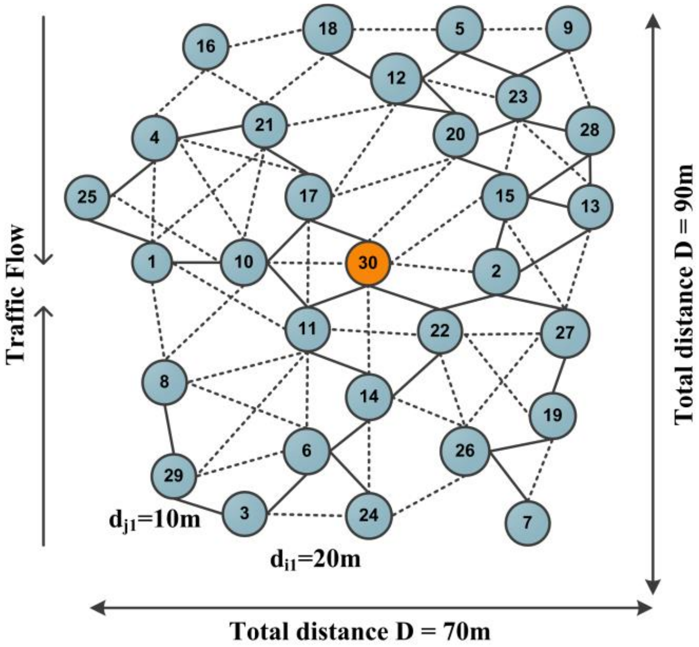


Figure 4.6 Hybrid deployment of nodes

The minimum distance between any given pair of nodes is 10 meters (solid line). If all the nodes employ T_x level 10, the network will be disconnected. This is because node 16 cannot be connected to rest of the nodes as its immediate neighbor is at 20m (dashed lines). Similarly, T_x levels 3 and 7 would not result in a connected graph. Therefore, the next T_x level that makes the network connected is 15. To get a general perception on the density of the nodes in various T_r, Figure 4.7 provides the T_r (solid arrow line) and I_r (dashed arrow line) range for node 22 for various T_x levels. Colored nodes are in interference range (I_r) and non-colored (T_r) are in transmission range of various T_x.

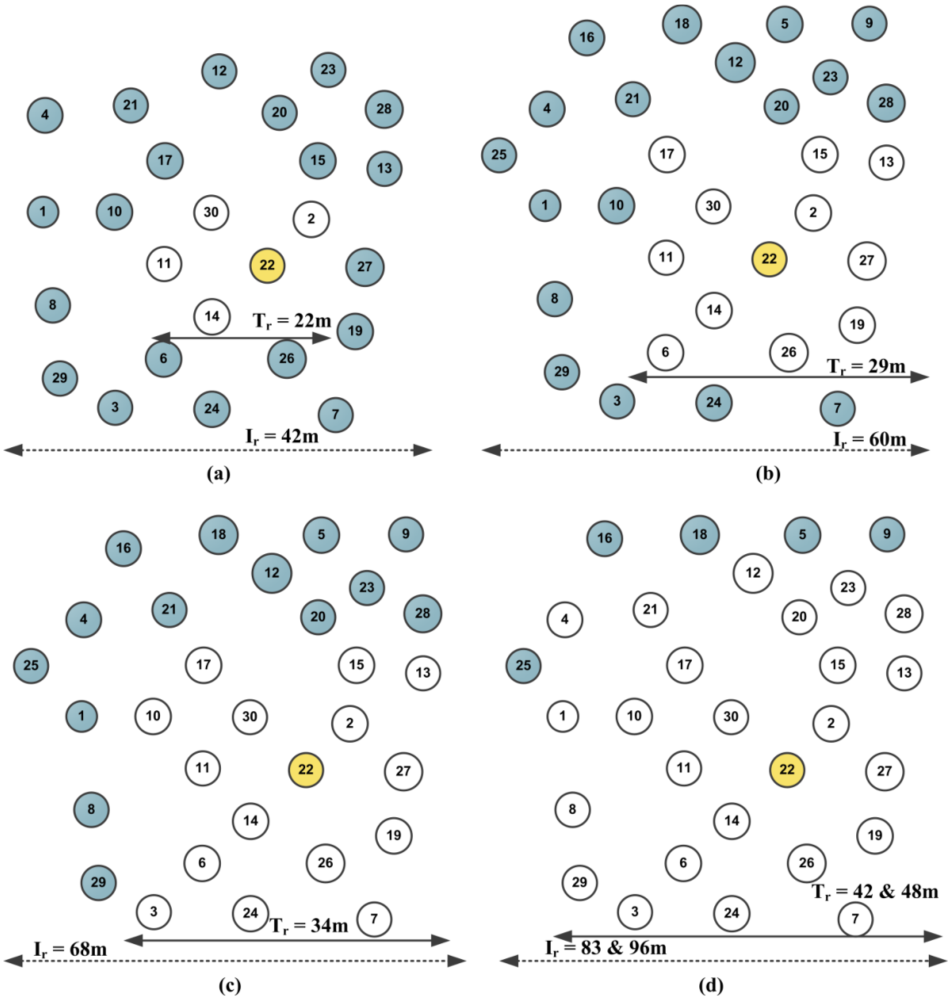


Figure 4.7 Transmission and interference range obtained for various power levels

4.4.3 Protocol Stack Configuration

At the application layer, all nodes are programmed to send data at three different rates –1 packet every 6, 12 secs and 1 packet randomly within 12 secs. We use RPL as routing protocols for both network deployments. RPL has gained popularity because it is known to perform well in Low Power and Lossy network (LLN). In addition, its IPv6 stateless auto configuration, with large address spacing makes it a worthy candidate for IoT network. Because of its high acceptance, RPL was a natural choice for our experiments. We chose

ContikiMAC because it is known to outperform its popular predecessors such as B-MAC and variants of X-MAC (X-MAC-C, X-MAC-CP, and X-MAC-P) in terms of latency, retransmission, energy consumption and PDR [97]. Finally, at the hardware level, all nodes use all the power levels except 3, 7 for linear deployment and 3, 7 and 11 for a hybrid deployment.

4.5 Results and Discussion

For each data rate (1 packet every 6, 12 secs and 1 packet randomly in 12 secs), routing (RPL) and MAC protocol (ContikiMAC), network deployment (linear and hybrid), the experiment is conducted for one T_x level. Next, the same experiment is re-run with different random seed and we increase the T_x to next level. This refers to one set of experiment. Therefore, in total, we have 18 and 15 experiments for linear deployment and hybrid deployment respectively.

The reliability of the network is represented by PDR that is calculated for end-to-end communication (source to sink). PDR is defined as a number of packets received by the sink divided by the total number of packets transmitted by the source node. Radio duty cycle metric provides the information on the percentage of time the transceiver was on for the entire duration of the experiment. Retransmission of packets provides the information about the collision in the network. In our experiments, by packets we mean the application data, control messages from routing protocol and the ACK packets produced by the 802.15.4 protocol on the reception of the packets.

4.5.1 Linear Deployment of 15 nodes

As T_x levels less than 11 (refer Table 4.1) does not connect the network, the experiment is started from T_x level 11. PDR for the network with various T_x levels and application data rate is shown in Figure 4.8.

As you can see that for T_x 11, the average PDR is the lowest, around 0.07 for almost all the data rates. As the T_x levels increase there is a drastic improvement in the PDR of the network. For T_x 31, we find that the average PDR is 1.

The reason for poor PDR in the case of lower power levels ($LowT_x$) is the hidden node terminal problem discussed in subsection 4.3.3. In addition, please note that the simulation uses UDGM radio model that emulates link failure due to fading of the signal as the node is farther away from the source. When all the nodes employ T_x level 11, it places the nodes at the very edge of the transmission range. This results in the formation of unstable links that has a high degree of failure. When the links fail, the routing protocol repairs the network by transmitting extra control messages. This extra flow of messages overwhelms the network resulting in less delivery of application data to the sink.

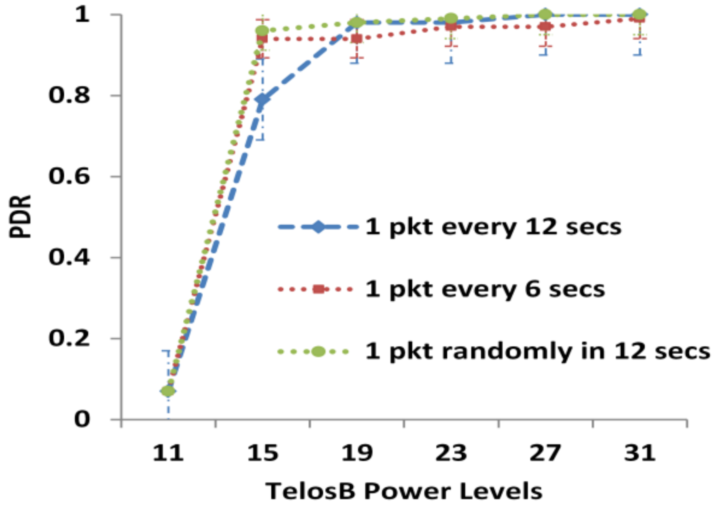


Figure 4.8 Average PDR of the entire linear network for various power levels and data rate

When the transmit power is high, the radio and its corresponding interference range are high. As a result, number of nodes is exposed to channel busy condition in a dense network and therefore they defer their communication. Hence, the chance of packets arriving at the destination increases. Furthermore, the quality of the link that is achieved with higher power levels is much more stable than the lower ones. Hence, the breakage of the links and thereby generation of control messages are lower. Table 4.2 provides the maximum (max); average (avg) and minimum (min) network PDR achieved for various data rates and T_x levels. Average PDR is for the entire network, whereas max and min PDR is for an individual node.

As you can see from Table 4.2 that for three different data rates, average PDR increases as the T_x levels increases. The same trend is also observed for max and min PDR. We also see that the variation in PDR is more for $LowT_x$ and they decrease as T_x rises. At lower T_x levels the Max PDR is 1 and it is because of the node that is closer (1-hop) to the sink. Except for the PDR of the nodes 13 and 14 that are closer to the sink, PDR of remaining nodes is very low.

Table 4.2 Max-Avg-Min PDR of the entire linear network for various data rate

Data Rates		**1 packet every 12 secs**			**1 packet every 6 secs**			**1 packet randomly in 12 secs**		
PDR		**max**	**avg**	**min**	**max**	**avg**	**min**	**max**	**avg**	**min**
Transmission Power (T_x) levels	11	1	0.07	0	1	0.07	0	1	0.07	0
	15	1	0.79	0.68	1	0.94	0.87	0.99	0.98	0.96
	19	1	0.98	0.94	1	0.94	0.87	1	0.98	0.96
	23	1	0.98	0.94	1	0.97	0.92	1	0.99	0.98
	27	1	1	0.97	1	0.97	0.89	1	1	0.90
	31	1	0.99	0.96	1	0.99	0.98	1	1	0.98

This indicates that for farther nodes (greater than 2-hops) at lower transmit power levels, the sink cannot receive data packets because of collision due to hidden node problem and because the low-power links are more susceptible to failure.

Retransmission occurs because of collision and it is evident in Figure 4.9 Collision is not solely because of multiple transmitting nodes, the main reason for it is $LowT_x$ that aggravates the hidden node problem. Failure of the low-powered link is more prominent at $LowT_x$. As the links break, routing protocol will try to find another alternative route to transmit the data to sink. However, this comes with the transmission of additional control messages. In addition, if the MAC protocol such as ContikiMAC has retransmission enabled, then more packets (both application data and control messages from routing layer) are retransmitted until ACK is received from the recipient [76].

In Figure 4.9, we find that as T_x level increases stepwise, there is a gradual linear reduction in a number of packets sent across at various data rate. For a data rate of one packet every 12 secs, the number of packets that are retransmitted for $HighT_x$ (31) is 28.5% less than the retransmissions at $LowT_x$ (11). Similarly, for a data rate of one packet every 6 secs; the retransmission is 21.32% less in the case of $HighT_x$ compared to $LowT_x$ (11). For the data rate of one packet sent randomly in 12 secs, the retransmission is 11% less in $HighT_x$ compared to $LowT_x$ (11). Compared to three different data rates, retransmission is the highest when nodes transmit one packet every 6 secs followed by one packet randomly sent in 12 secs. The reason for this lies in the frequency of data transmitted.

Higher the frequency, higher is the rate of collision (more prominent at $LowT_x$) and hence more time is required by the MAC layer of the pair of nodes to synchronize its communication.

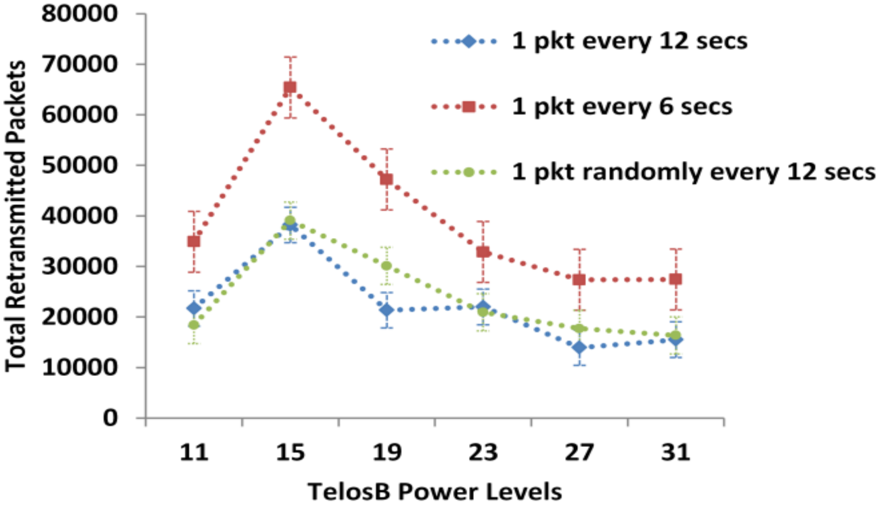


Figure 4.9 Retransmitted packets in the entire linear network for various power levels and data rate

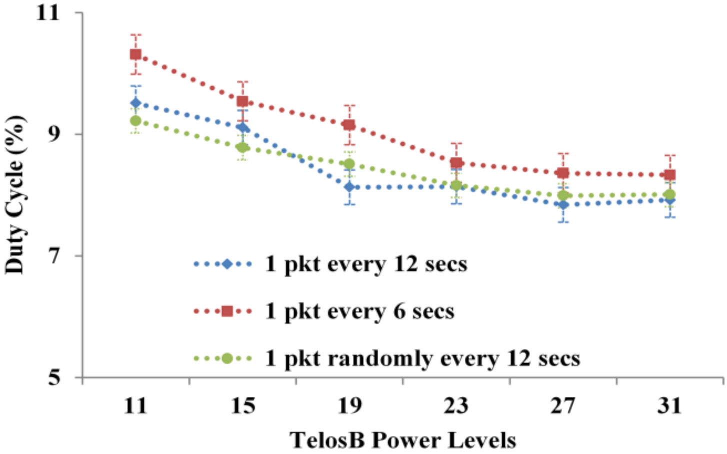


Figure 4.10 Duty cycle of the entire linear network for various power levels and data rate

When the data rate is random, the MAC protocol of the receiving nodes cannot easily synchronize its sleep-wake up time interval according to the transmission from intermediate nodes.

It is desired to have high PDR. However, the cost in terms of energy to achieve it must be minimal. We expect that the energy consumption of the network to be less at $LowT_x$. However, this is not the case as we find in Figure 4.10. This is because as the number of retransmission increases the duty cycle of the nodes eventually increases. From Figure 4.10 we find that the duty cycle of nodes while transmitting with $HighT_x$ is less by 16.72%, 19.20%, and 13.12% compared to $LowT_x$ for one packet sent every 12, 6 and one packets randomly sent in 12 secs respectively. The retransmission is more with lower power levels as hidden node problems cause the MAC protocols to retransmit more packets. As the nodes are kept in transmit and listen mode for a longer duration at a higher data rate (6 secs), the duty cycle being elevated is expected.

4.5.2 Hybrid Deployment of 15 nodes

If we refer to the Figure 4.6 in subsection 4.4.2, we find that employing T_x levels below 15 would not result in a connected graph and therefore the experiments for 30 nodes circular deployment is started from T_x level 15. The perception that PDR of relatively large network falls due to a collision at $HighT_x$ is false. This is evident in Figure 4.11 that provides PDR of the network for various data rates. Compared to $MaxT_x$ (31), PDR of the network at $LowT_x$ (15) is less by 37%, 63%, and 31% for one packet sent every 12, 6 and 1 packet randomly in 12 secs respectively. The reason for this is the same as we provided in the previous section. The lowest T_x for linear deployment is 11, whereas for hybrid deployment it is 15. Compared to PDR at $T_{x,}$ level 11 for linear deployment, the PDR at T_x level 15 for hybrid deployment is 88%, 80% and 89% more for a packet sent at 12, 6 and one packet randomly sent in 12secs respectively. This is because the number of nodes at 1– hop distance from the sink that can be reached with $LowT_x$ (15) in hybrid deployment is more than its counterpart.

Table 4.3 provides the max-avg-min network PDR range for various data rates. If we refer to the max-avg-min table of the linear deployment, we find that in linear deployment the variation reduces from T_x level 15. However, this is not the case with a hybrid deployment.

The variation shortens after T_x level 23 for all the data rates. This is because the network is large and to mitigate collision due to the hidden-node problem, higher transmit power is required. At higher transmit power, more nodes fall in the transmission range, and hence more nodes will defer their communication causing lesser deviation in PDR and lesser retransmission of packets.

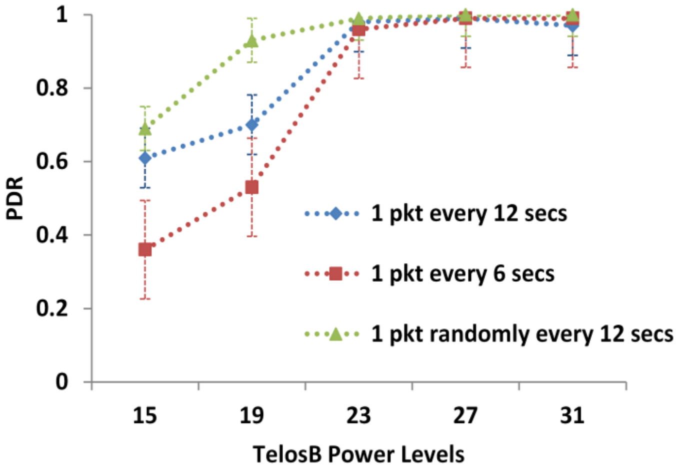


Figure 4.11 Average PDR of the entire hybrid network for various power levels and data rates

Table 4.3 Max-Avg-Min PDR of the hybrid network for various data rate

Data Rates		1 packet every 12 secs			1 packet every 6 secs			1 packet randomly in 12 secs		
PDR		max	avg	min	max	avg	min	max	avg	min
Transmission Power (T_x) levels	15	0.92	0.61	0	0.68	0.36	0	0.95	0.69	0
	19	0.95	0.7	0.32	0.77	0.55	0.39	0.99	0.93	0.82
	23	1	0.98	0.87	1	0.96	0.49	1	0.99	0.92
	27	1	0.99	0.94	1	0.99	0.98	1	1	0.99
	31	1	0.97	0.7	1	0.99	0.94	1	1	0.99

The trend with retransmitted packets for hybrid deployment as shown in Figure 4.12 is similar to that of linear deployment. Compared to the retransmission at $LowT_x$ (15), the retransmission at $HighT_x$ (31) for one packet sent every 12, 6 and 1 packet randomly sent in 12 secs is less by 89%, 93%, and 89% respectively.

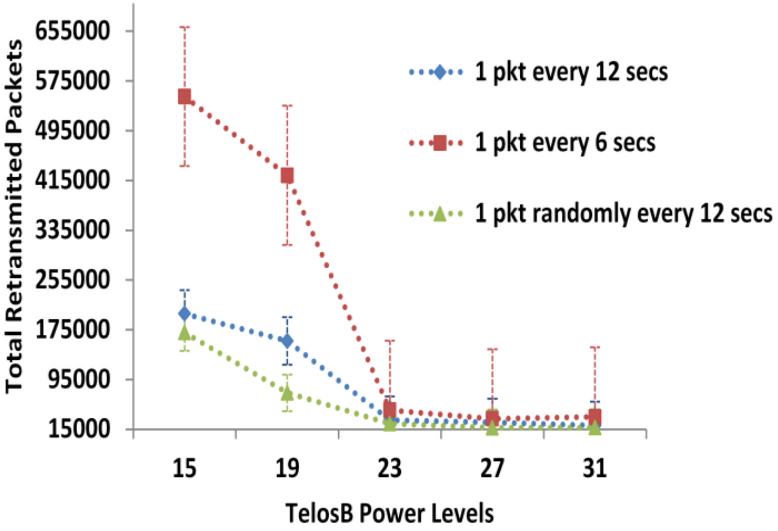


Figure 4.12 Retransmitted packets in the entire hybrid network for various power levels and data rate

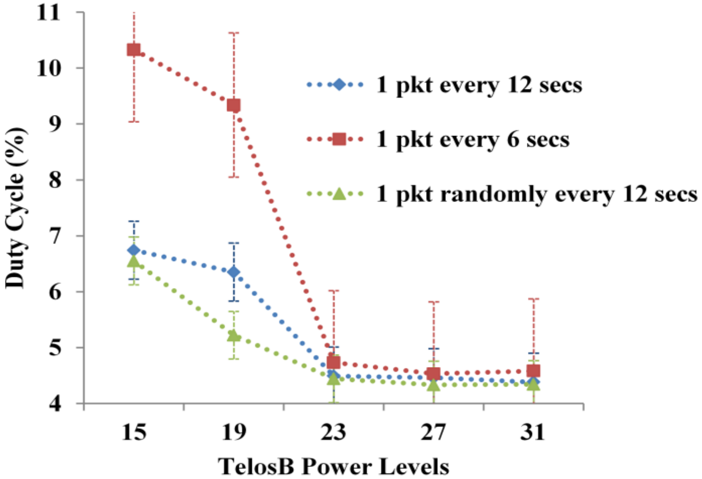


Figure 4.13 Duty cycle of the entire hybrid network for various power levels and data rate

If we observe the Figure 4.12 and Figure 4.13 depicting a number of retransmitted packets and the duty cycle for the hybrid network, we find that at higher transmit power they decrease substantially. We find that the duty cycle of nodes while transmitting with $HighT_x$ (31) is less by 35%, 55% and 33%

compared to $LowT_x$ (15) for 1 packet every 12, 6 and 1 packet randomly in 12 secs respectively. In both deployments, at higher transmit power levels; the duty cycle is lesser for random data rates.

This is a desirable property to have, as most of the IoT networks are event-based. By transmitting only data of interest, the network will consume less energy. However, it is necessary to study how MAC protocol in the pair of nodes can synchronize its sleep-wake time interval at random data rate. Our study shows that even at higher transmit power levels, retransmission in the network, the duty cycle of the nodes does not increase.

Figure 4.14 and Figure 4.15 shows the power consumption of the transceiver for different deployment scenarios. The drop in the energy consumption is in line with the duty cycle of the nodes for different deployment scenarios.

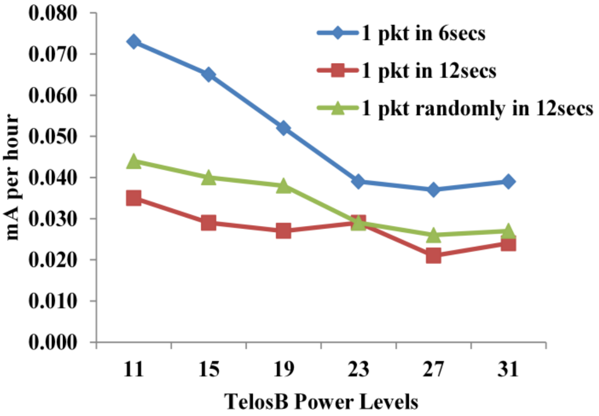


Figure 4.14 Power consumption of all the transceivers for linear deployment

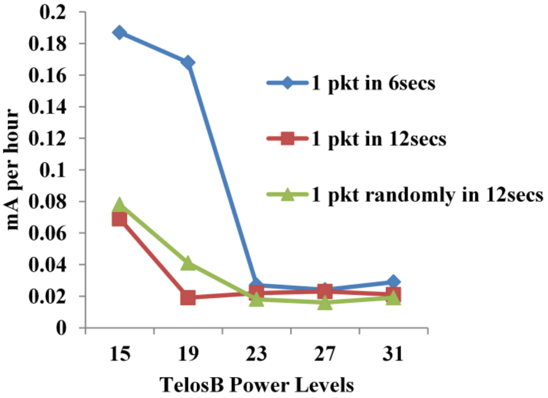


Figure 4.15 Power consumption of all the transceivers for hybrid deployment

4.6 Conclusion and Future Work

In this chapter, we studied the performance of the networks in terms of PDR, duty cycle and collision rate for various data rate and transmission powers. The main research objective of this chapter was to investigate if there is a trade-off in terms of reliability and energy consumption between $LowT_x$ and $HighT_x$. Our experiments show that low power links are more susceptible to failures than high power ones. $LowT_x$ causes more collision in the network because of the hidden-node problem. The general misconception is that the sensor nodes employing $HighT_x$ increase the chances of collision, resulting in higher retransmission and energy utilization. However, we observed that compared to $HighT_x$, $LowT_x$ fares poorly across all the metrics. This suggests that node employing T_x that places most of the nodes in its transmission range is the optimal T_x as it thwarts hidden-node problem in a dense network with multiple transmitting nodes. The routing and MAC protocol used was RPL and ContikiMAC. In the future one may consider investigating the performance of the network with other popular routing protocols such as Ad-Hoc On Demand Distance Vector (AODV), Collection Tree Protocol (CTP) etc. and MAC protocols such as B-MAC, X-MAC etc. at various power levels and data rate.

Chapter 5 Assessment of Non-Machine Learning Algorithms for Transmission Power Control

The existing TPC schemes adjust the transmission power (T_x) mostly reacting to changes in link quality between communicating nodes. Proactive TPC has been proposed in the recent past, as reactive TPC does not predict the fall in the link quality between a pair of nodes. Early prediction of link quality can increase the reliability of the link between the nodes as T_x can be adjusted accordingly beforehand to compensate the degradation in the propagation medium. Although popular machine learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN) etc. could be used for link quality prediction, they require a large amount of training samples for training them before they are capable of predicting. In addition, implementing them in popular embedded IoT devices such as TelosB, MicaZ, and Econotag etc. is challenging due to their hardware limitations.

Considering the time required for collecting the data, training the learning model and hardware limitation of IoT devices, learning algorithms are not the best choice. Therefore, the main objective of this chapter is to investigate other alternatives to learning algorithms. To this end, this chapter addresses the following research objective mentioned in chapter 1

(RO4) *Which other non-machine learning algorithms can predict the link variation?*

This chapter provides a comprehensive analysis of prediction accuracy and ease of configuration of some of the many non-machine learning algorithms such as Discrete Kalman Filter (DKF), Exponentially Weighted Moving Average (EWMA), Simple Moving Average (SMA), and Weighted Moving Average (WMA) that could be employed in a proactive TPC technique. These algorithms are chosen because they can be easily implemented in resource constraint sensor nodes. In addition, they do not require tweaking of multiple parameters. Linear Regression (LR) is used to show a number of training samples required by a batch based machine-learning algorithms.

Experiments indicate that the prediction accuracy of DKF has the least forecasting error and outperforms the prediction accuracy of all other algorithms under discussion. Amongst moving average algorithms, the prediction accuracy of WMA is significantly better and linear regression algorithm has the worst performance. Evaluating the cost involved in terms of operational uptime, ease of configuration and implementation, WMA is the best algorithm for implementing proactive TPC.

The rest of the chapter is organized as follows. Section 5.1 highlights the demerits of employing machine-learning algorithms for TPC. Section 5.2 provides the information about the experimental setup. Section 5.3 presents the characteristics of input data. Section 5.4 briefly introduces the algorithms evaluated. Section 5.5 provides the information about the evaluation metrics. Section 5.6 provides the performance evaluation in terms of prediction accuracy of the algorithms in the discussion. In Section 5.7, the conclusion is provided.

5.1 Drawbacks of Employing Machine-Learning for TPC

Supervised learning has two phases - Data collection and training phase. In the former phase, data (link quality of communication medium) is collected over a period. For instance, learning algorithms such as Locally Weighted Projection Regression requires a large number of data samples to make a decent prediction [116]. In the latter phase, the learning model is trained to identify low, average, or good link quality value. These two phases are time-consuming processes and given the energy constraint of the IoT devices, supervised learning does not offer a scalable solution for predicting the link quality in a short period in a dense network.

The major demerit of many IoT devices currently used in a sensor network is the unavailability of floating point math units. Learning algorithms such as SVM, Logistic regression requires floating math units for their computation [63]. To overcome this disadvantage, fixed-point math is used. However, usage of fixed point math has drawbacks in terms of overflow and underflow with multiplication and division operations that could lead to error in prediction [117]. An algorithm such as Naïve Bayes requires each training instance to be visited and each of its features counted for predicting every new data points. This feature of the algorithm delays the prediction capability and puts a load on the memory requirement. Many of the network-embedded devices have as little as 10kb RAM.

The values of link quality metrics such as Received Signal Strength Indicator (RSSI) are non-linear in nature. The usage of an algorithm such as Naïve Bayes and SVM with linear mapping function may result in inferior prediction [118][84]. Table 5.1 highlights some of the drawbacks of prominent learning algorithms that are used in Wireless Sensor Network.

Table 5.1 Drawback of popular machine learning algorithms

Machine Learning Algorithms	Drawbacks
Locally Weighted Project Regression	Requires large data samples
Logistic Regression	Requires vector multiplication and sigmoid function calculation. This is not feasible in resource constraint device [63]
Support Vector Machine	Selection of suitable kernel, Parameter values, and appropriate error cost is difficult [119]
Naïve Bayes	Each training instance needs to be visited and each of its features counted for prediction
Neural Networks	High computational cost for learning makes it non pervasive in distributed environment such as WSN [120]
Decision Tree	Performs well only with linearly separable data [120]

5.2 Experimental Set-up

In order to check the accuracy of prediction algorithms, the link quality data collected from the spatially varying real-world scenarios were used. The experiments were performed using TelosB motes having CC2420 radio chipset in two different real-world conditions such as [16]

- Connected Region: Here the link quality between the sensors is often good and stable.
- Transitional Region: In this region, link quality between the sensors is very unstable and hence unreliable.

The extent of these regions depends mainly on the environment - indoor and outdoor [121]. Hence, the connected and transitional regions for experiments were achieved by varying the distance between the transmitting (T_x) and receiving (R_x) nodes in indoor and outdoor settings. The deployments were in actual office and street environments with high human mobility.

Figure 5.1 represents the system setup that was used for the experiment. The data from T_x is received by R_x and transferred to a laptop via USB port. The laptop contains various algorithms that forecast the quality of the link. Table 5.2 provides the distance between T_x and R_x for various scenarios.

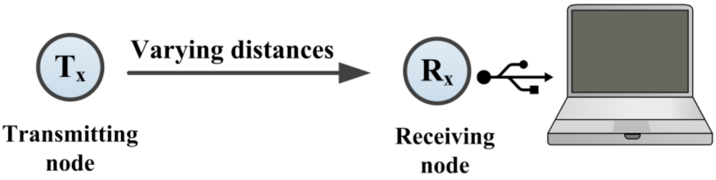


Figure 5.1 Experimental setup

For the transitional indoor region (scenario 1), the distance between T_x and R_x was set to 11.50 meters. For the transitional outdoor region (scenario 2), the distance between T_x and R_x was set to 18.70 meters.

In the connected indoor region (scenario 3), the distance between T_x and R_x was set to 5 meters. The distance between T_x and R_x in the connected outdoor region (scenario 4), was set to 8.7 meters respectively.

Table 5.2 Distance of Tx from Rx

Region	Distance from T_x from R_x
Transitional Outdoor (scenario 1)	18.70m
Transitional Indoor (scenario 2)	11.50m
Connected Indoor (scenario 3)	5m
Connected Outdoor (scenario 4)	8.7m

The distances between the sensors from each other were chosen randomly to match the realistic WSN deployment. This provided us the platform to check the prediction accuracy of the selected algorithms in varying conditions. Finally, following settings were kept constant for the entire experiments

1. For every one second, T_x node was configured to send 28 bytes of data to R_x on the default channel 26 as specified by CC2420.

2. Default maximum transmission power level of 31 as specified by CC2420 was used by the T_x. This power level corresponds to 0dBm.
3. The battery power of T_x and R_x node was set to 3V.
4. Once the position of T_x and R_x was set, their position was not altered during the course of nodes sending the packets. Thus, the sensor network was static.
5. Algorithms under discussion were run on normal laptops off-line.

We used RSSI for analyzing the link quality between the nodes. In CC2420, the RSSI is calculated over eight symbols and stores the result in RSSI.RSSI_VAL register [122]. Texas Instrument uses the following equation 5.1 to calculate the received signal power in dBm.

$$P = RSSI\_VAL + RSSI\_OFFSET \tag{5.1}$$

Empirically it is found that *RSSI_OFFSET* value is set to -45 dBm [28]. *RSSI_VAL* refers to the obtained RSSI value. Experiments show that LQI must be averaged over many samples before it can really estimate the link quality [16]. This is also the case with Packet Delivery Ratio (PDR). Therefore, RSSI was used as an input for algorithms under discussion.

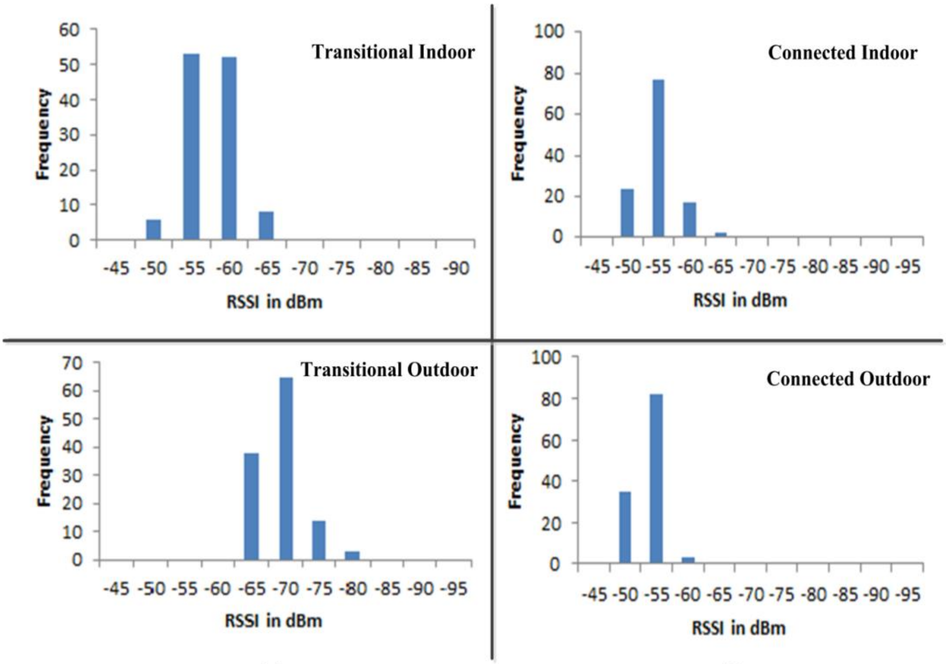
5.3 Characteristics of Input Data

Figure 5.2 provides the histogram of received signal strength index (bin size of 5 dBm) derived from 120 packets transmitted over two minutes.

Table 5.3 Standard deviation and variance of RSSI values

Region	Standard Deviation	Variation
Transitional Outdoor	2.62	6.86
Transitional Indoor	3.4	12.24
Connected Indoor	2.7	7.5
Connected Outdoor	2.0	4.5

From Figure 5.2(a), Figure 5.2(b) and Table 5.3, we can infer that the variation in RSSI is significant for at least two regions –Transitional Indoor and Outdoor. This is an indication that the channel condition can vary instantly and there is a genuine need to forecast the link quality and adjust the power level to ensure good communication reliability.



(a) (b)

Figure 5.2 Frequency distribution of RSSI values in various scenarios

5.4 Overview of Algorithms Evaluated

In this section, we briefly explain five algorithms that can be used in TPC. All the methods below use collected RSSI values X_i, to forecast the next RSSI value $\widehat{X}_t$, at any time slot t.

1. Simple Moving Average (SMA) [21]**:** Is the unweighted mean of the last N RSSI values. The next RSSI value is forecasted by summing the series of past RSSI values as shown in equation 5.2. X_{t-i} is the past RSSI value at time $t-i$ where $1 \leq i \leq N$.

$$\hat{X}_t = \frac{X_{t-1} + X_{t-2} + X_{t-3} + \ldots X_{t-N}}{N} \tag{5.2}$$

2. Weighted Moving Average (WMA): Applies weight to all past *N* RSSI linearly. This implies that the latest RSSI values X_{t-1}, X_{t-2} …, X_{t-N} are given more weightage and has the impact that is more significant on the average value than the previous RSSI values. The equation for WMA is shown in equation 5.3.

$$X_t = \frac{N * X_{t-1} + (N-1) * X_{t-2} + \ldots + X_{t-N}}{N + (N-1) + \ldots + 1} \tag{5.3}$$

3. Exponential Weighted Moving Average (EWMA) [123]**:** Similar to WMA, as shown in equation 5.4, EWMA is a weighted average of the last *N* RSSI values. However, the weight decreases exponentially with each incoming RSSI.

$$X_t = \alpha(X_{t-1}) + (1-\alpha) S_{t-1} \tag{5.4}$$

Here, $\widehat{X_t}$ is the value of RSSI at time slot *t*, S_t is the exponential moving average at time *t*. α is a smoothing factor and can take values between 0≤α≤1. By choosing appropriate α value, EWMA can be made sensitive to a small variation in RSSI.

4. Linear Regression (LR) [124] is used for modelling the relationship between dependent variables (RSSI values X_{t-1}) and an independent variable (time). Mathematically, LR is represented as shown in equation 5.5

$$\hat{X}_t = \beta_1 X_{t-1} + \ldots + \beta_N X_{t-N} + \varepsilon_i \text{ ; i=1,2,..N} \tag{5.5}$$

5. Kalman Filter (KL) [125]**:** For static wireless network, the value of RSSI was modelled as shown in equation 5.6 [83]

$$X_t = X_{t-1} + W_{t-1} \tag{5.6}$$

$$\widehat{Z_t} = \widehat{X_t} + V_t$$

Where $\widehat{X_t}$ represents the RSSI at time slot *t* and $\widehat{Z_t}$ is the RSSI measurement calculated when the packet is received at time slot *t*. The noise in the process of X_t and the measurement noise in Z_t is modelled as Gaussian processes W_{t-1}~ N (0, Q) and V_t ~N (0, R) respectively. The value of Q is the variance of the RSSI values of the broadcast messages sent by sensor nodes during the initialization process. R is the variance of the measurement noise in dBm calculated by the sensor nodes before the transmission.

5.5 Performance Evaluation Metrics

As discussed earlier, WSN does not operate in a deterministic environment [17]. Therefore, the algorithms used in proactive TPC must be robust enough to forecast the behavior of the link quality. In order to evaluate the prediction accuracy of the algorithms mentioned in section 5.4, following statistical formulas are used.

1. Mean Absolute Percentage Error (MAPE): MAPE is the summation of the absolute difference of forecasted values $\widehat{X}_t$ and eventual outcomes X_i divided by a total number of N RSSI values. The value of MAPE ranges from zero to infinity. A predictor having MAPE value of zero is considered an ideal prediction algorithm. MAPE is represented as shown in equation 5.7

$$\frac{100}{N}\sum_{t=1}^{N}\left|\frac{X_t - X_t}{X_t}\right| \tag{5.7}$$

2. Mean Forecast Error (MFE): MFE is an indicator of forecasting bias and is calculated as shown in equation 5.8

$$\sum_{t=1}^{N}\frac{X_t - X_t}{N} \tag{5.8}$$

An ideal MFE would be zero. If MFE is greater than zero, it indicates that the prediction algorithm has under-forecasted and if MFE is less than zero, the prediction algorithm has over-forecasted.

3. Root Mean Square Error (RMSE): Calculates the standard deviation of the differences between the observed values X_t and predicted values $\widehat{X}_t$. It is calculated as shown in equation 5.9

$$\sqrt{\sum_{t=1}^{N}\frac{(X_t - X_t)^2}{N}} \tag{5.9}$$

Large errors in the forecast have more impact on the value of RMSE than the small errors. The value of RMSE range between zero to infinity. The estimating algorithm having an RMSE of value zero is known to be ideal. The assessed algorithms were configured as follows

1. *N* is initialized as the total number of RSSI values of each data packets minus the total number of training RSSI values of corresponding data packets.
2. The performance of WMA is evaluated by setting the weight to 2 and 3.
3. The performance of EWMA is evaluated by setting α to 0.9 and 0.6.
4. β the regression coefficient for the linear regression was calculated from the RSSI values corresponding to 30 and 20 data packets respectively.

5.6 Evaluation of Algorithms

5.6.1 Performance of SMA and WMA

Table 5.4 provides the prediction accuracy of SMA and WMA in four different scenarios. Two periods and three periods in the Tables 5.4 and 5.5 represent the number of RSSI values of corresponding data packets required by the algorithms before predicting the subsequent RSSI values. (Periods also refer to the training data sample).

Table 5.4 Prediction errors of SMA and WMA

Algorithms	Accuracy Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
2 period SMA	MAPE	2.04	1.28	3.68	3.11
	MFE	-0.04	0.09	-0.06	-0.03
	RMSE	1.81	1.46	2.95	2.34
2 period WMA	MAPE	2.05	1.21	3.74	3.03
	MFE	-0.04	0.08	-0.06	-0.02
	RMSE	1.86	1.40	2.94	2.32
3 period SMA	MAPE	1.99	1.37	0.16	3.23
	MFE	-0.05	0.12	-0.06	-0.04
	RMSE	1.77	1.56	2.74	2.38
3 period WMA	MAPE	1.99	1.29	3.47	3.08
	MFE	-0.04	0.10	-0.05	-0.03
	RMSE	1.79	1.45	2.78	2.31

From Table 5.4, we find that the statistical value of metrics of 3 periods WMA is better than its counterparts in various spatially varying scenarios.

These statistical values indicate that the forecasting accuracy of the moving average algorithms in TPC can be enhanced by increasing the number of periods and by providing more weightage to the latest RSSI values.

However, increasing the number of periods means all the sensor nodes must perform longer initialization phase wherein each sensor node broadcast packets equal to the number of periods before the actual communication. This indicates that network will spend more time in initialization phase transmitting additional probe packets and hence there is an increase in the power consumption. Therefore, there is a trade-off between increasing the prediction accuracy and reducing the transmission cost.

5.6.2 Performance of EWMA

Table 5.5 provides the comparison between EWMA with different smoothing factor (α) values for various scenarios. As we can understand from Table 5.5, increasing the smoothing factor α does not necessarily improve the prediction accuracy. The reason for this is that if we give α value closer to one, more weightage is given only to the recent dataset. The choice of having appropriate smoothing factor is often a difficult task and it determines the accuracy of EWMA. Necessary details to choose an appropriate smoothing factor is provided in [126].

Table 5.5 Prediction errors of EWMA

Algorithms	Accuracy Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
2 period EWMA (α=0.6)	MAPE	2.06	1.28	2.96	3.68
	MFE	-0.07	0.09	-0.03	-0.06
	RMSE	1.84	0.30	2.24	2.95
2 period EWMA (α=0.94)	MAPE	2.2	1.15	3.04	3.99
	MFE	-0.03	0.06	-0.01	-0.04
	RMSE	2.08	1.37	2.43	3.12

Like SMA and WMA, EWMA with smoothing factors 0.6 and 0.94 utilizes a minimal amount of packets during the initialization phase. However, the prediction accuracy of two periods EWMA with α=0.94 and α=0.6 is not good compared to both two and three periods SMA and WMA. Obtaining appropriate α value is not straightforward and needs more trial and error approach. Hence, when it comes to ease of configuration, EWMA falls short. One way to increase the prediction accuracy is to increase period size. However, this will result in a prolonged initialization phase with an uptick in power consumption.

5.6.3 Performance of Linear Regression

From Table 5.6, we can infer that the linear regression has the worst performance. While testing the prediction accuracy of linear regression, we found that considerable amount of RSSI values are required (more than 30 packets) before predicting future RSSI values with minimum statistical errors. As shown in Table 5.7, any packets less than 30, significantly increases the statistical errors.

Table 5.6 Prediction errors of linear regression with β calculated from 30 RSSI values

Algorithms	Accuracy Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Linear Regression	MAPE	12.70	10.01	8.51	12.82
	MFE	7.80	-6.45	4.55	6.88
	RMSE	8.78	8.61	5.36	8.19

Table 5.7 Prediction errors of linear regression with β calculated from 20 RSSI values

Algorithms	Accuracy Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Linear Regression	MAPE	20.57	12.25	28.05	14.90
	MFE	12.71	-8.44	15.91	8.16
	RMSE	14.53	10.87	17.52	9.72

The reason behind the bad performance lies in fitting the line to already available RSSI values. In linear regression, the slope that is used to fit the data points is not updated as new data arrive. In order to improve the accuracy, we need a mechanism such as a sliding window that aids in updating the slope, as new data points are available [127].

5.6.4 Performance of Kalman Filter

From Table 5.8 we can conclude that Discrete Kalman filter gives the best accuracy in different realistic scenarios when compared to all the algorithms discussed. Three packets were made available (3 periods) to Kalman filter before predicting the future RSSI values. To get the accurate prediction as shown in Table 5.8, every node must calculate the variance in noise. To calculate the variance in the noise, R_x node must be in listening mode for the longer duration. Many IoT devices such as TelosB, MicaZ, and Econotag consume more energy in listen-mode than in transmit-mode. This could be considered as a demerit of using Kalman filter.

Table 5.8 Prediction errors of Kalman Filter

Algorithms	Accuracy Metrics	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Discrete Kalman Filter	MAPE	0.80	0.10	3.73	3.53
	MFE	-0.03	0.02	-0.11	-0.09
	RMSE	0.91	0.26	3.26	2.62

5.7 Conclusions

In this chapter, drawbacks of employing supervised learning in resource constraint wireless embedded devices were highlighted. One of the drawbacks is related to the time-consuming process of gathering data and training the prediction model before they become operational. The second drawback is associated with the hardware limitations.

As non-machine learning algorithms such as variants of moving average and discrete Kalman filter are easy to configure and implement in low-power embedded devices, these algorithms were chosen and their prediction accuracy was evaluated in the spatially varying realistic environment. Testing in four different environmental settings, we found that Discrete Kalman Filter has the best accuracy. Although Kalman Filter has best accurate in finding the future link quality with minimum communication and configuration overhead, it needs the variance in the noise floor that can be obtained from SNR. Computing SNR involves a high cost in terms of communication power because the radio must be kept in listening mode for prolonged period.

The accuracy of WMA is the second best followed by SMA. Although their accuracy is lower than that of Discrete Kalman Filter it can be improved by marginally increasing the packets (two or three packets) broadcasted in the initialization phase.

EWMA is the third best algorithm in terms of forecasting accuracy because it does not outperform SMA and WMA. The smoothing factor α plays a crucial role in the efficiency of EWMA and configuring it is not a straightforward approach. Hence, the network must be put through a testing phase before it is deployed to get appropriate α value.

As RSSI variation is not linear, linear regression has the worst performance compared to the other algorithms. Although the regression model is provided with ten times more data points (RSSI values) during initialization phase compared to other algorithms, the least square approach employed in regression to construct the line that fits the data fails. Employing non-linear regression model such as polynomial regression may be considered in the future to predict the link quality.

As discrete Kalman filter employs SNR that requires nodes to spend extra energy as they are kept in the listen-mode more than default case and given the prolonged configuration phase involved in enhancing the prediction accuracy of EWMA, We find WMA to be the optimal algorithm to be utilized in proactive TPC for resource constraint sensor nodes.

Chapter 6 Data Aware Transmission Power Control (DA-TPC)

Reactive TPC algorithms such as ODTPC, AODPTC collect network metrics such as RSSI, LQI, and PDR for certain time window *w*. Based on the deviation of the specified metric from a predefined threshold level, transmission power (T_x) is recommended by the receiving node. As discussed in chapter 3, this reactive model of TPC is known to impede the network performance with a static routing protocol such as ContikiMesh. In addition, the performance of the individual node involved in TPC is also below average. Although reactive TPC with a dynamic routing protocol such as Collection Tree Protocol (CTP) at lower layer performs marginally better at the individual node level, it consumes more energy compared to ContikiMesh.

Furthermore, metrics such as RSSI, LQI that are utilized as an input for TPC algorithms are sensitive to environmental conditions. Therefore, TPC algorithms to learn the variation in environmental condition perform a lengthy and energy inefficient initialization phase. Employing these sensitive parameters tightly couples the TPC to specific conditions and hence lot of trial and error phase is needed to make the TPC work when the network is deployed to a new location. This makes TPC algorithms non-generic.

TPC algorithms proposed in the literature employ T_x level just enough to connect to an immediate neighbor. These low power links are prone to failure due to internal interference. In addition, it was experimentally shown in chapter 4 that when an individual node in the network selfishly choses a low transmission power ($LowT_x$) it alleviates the hidden-node problem. This causes a collision in the network when there multiple transmitting nodes and hence more packets are retransmitted at the expense of higher energy consumption. Keeping these relevant problems in mind, this chapter addresses the following research objective mentioned in chapter 1

(RO5) *Which another alternative metric can be used by the TPC algorithms to change the* T_x*?*

To this end, this chapter proposes a novel sender-side Data Aware Transmission Power Control (DA-TPC) algorithm that uses priority of the data rather than sensitive parameters such as RSSI as the decision point to adjust the T_x. DA-TPC has a shorter initialization phase and selects appropriate T_x that reduce the hidden-node problem. Experiments conducted show that the nodes employing DA-TPC perform better than reactive TPC in terms of Packet Delivery Ratio (PDR), latency and energy consumption. In addition, the results also reveal that frequent fluctuation of T_x by DA-TPC does not affect the performance of the entire network as it does in the case of reactive TPC.

In section 6.1 drawbacks of TPC model proposed in the literature is outlined. Overview of the DA-TPC algorithm is outlined in section 6.2. Various components of DA-TPC are presented in section 6.3. Section 6.4 provides information on experimental setup. Results are discussed in section 6.5. Finally, conclusion and future work is outlined in section 6.6

6.1 Drawbacks of TPC Model

The general working of TPC algorithms discussed in the literature is elaborated in Figure 6.1. In Figure 6.1, all the nodes N; where N=$n1$, $n2$... $n5$ in the network perform initialization phase.

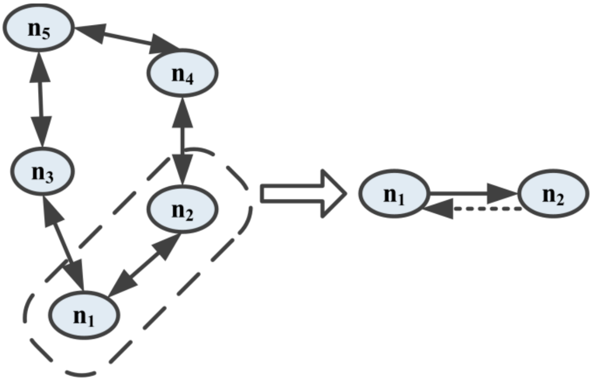


Figure 6.1 Receiver side link quality based TPC design

Consider a subset of the network that consists a pair of nodes ($n1$, $n2$) encircled with dashed line. The node $n1$ broadcasts a certain amount of probe messages to the receiver node $n2$ using different transmission power (T_x) levels supported by specific radio transceiver of the sensor node. Node $n2$ then builds a model that reflects the correlation between the transmission power and the link quality. Based on the knowledge inferred from the model, node $n2$ recommends specific power level to node $n1$ (shown by dashed lines in Figure 6.1) to use for future communication in case there is a variation in the link quality in course of time. This generic TPC design has following drawbacks

1. Building a model on the receiver node that accurately correlates the T_x with link quality consumes high T_x. This is because a significant amount of probe packets has to be sent from the sender node. Furthermore, initialization performed by all the nodes at different time slots results in the delay the network becomes fully operational [57]. Experiments show that in some environments, the even long initialization phase is not sufficient to predict the behavior of link quality between the pair of nodes [33]. ATPC, PCBL are some of the TPC algorithms that employ prolonged initialization phase [68][54].
2. As there is overhead involved in the initialization phase, some of the TPC algorithms such as ART, P-TPC skips the initialization phase and directly employ PDR metric of the actual transmitted data to recommend the change in the T_x level. However, substantial amount PDR values are required to estimate the link quality [79]. This indicates that reaction to change in the link quality is slow and reactive.
3. As a substantial amount of PDR values are needed to estimate the link quality, TPC algorithm such as ODTPC employs RSSI to assess the link quality on per packet basis [57]. However, adopting it as a network metric for the initialization phase is more error prone [46]. This is especially the case when all the nodes simultaneously perform the initialization phase to reduce the time the network actually becomes operational to transmit sensor data. RSSI is the summation of the signal strength and the noise floor. The noise floor is the transmission signal from another adjacent pair of nodes that are not directly in communication. To calculate noise floor the nodes must be in energy inefficient listen-mode for a longer duration. Therefore, utilization of RSSI as a metric by the receiver nodes to calculate the T_x level to be adopted by the sender nodes is not appropriate.
4. TPC algorithms such as ODTPC, MODTPC recommends a T_x to its respective neighbors that are just enough to maintain a communication link between them [128]. To conserve the energy this seems to be a valid solution only when every node in the network is scheduled to send their respective data at fixed time slots to avoid interference. However, when the network comprises of multiple source nodes transmitting data simultaneously, the link established with bare minimum T_x tend to break more often (refer chapter 3). This results in the frequent adjustment of T_x more often than required causing unstable

network. In addition, selection of $LowT_x$ by the nodes gives rise to hidden-node problem (refer chapter 4).

5. As a part of an effort to maintain certain network topology, prominent routing protocols such as RPL, AODV, and CTP transmit control messages quite often. As extensively discussed in chapter 3, a number of control messages can increase substantially when there is a frequent adjustment of T_x causing instability in the network [35]. There is a potential risk of disrupting previously well-connected network due to the collision. As a result, routing protocols may take extra time and energy to readjust and find a new best route.

6. A receiver based TPC (TPC that employ RSSI, LQI, SNR, PDR, ETX metrics) consume more energy because the receiver node must recommend appropriate T_x to the source node. Control messages from the routing layer along with frequent piggybacking of the T_x level recommendation from the receiver node can drain the battery of the nodes sooner.

7. All the metrics (e.g. RSSI, LQI, SNR) discussed so far are either sensitive to an environmental condition or employing them as an input to TPC makes them slow and reactive to changes in a propagation medium (e.g. PDR, ETX). Due to the uniqueness of a location where the IoT network is deployed, link quality threshold level may differ. As a result, calibrating them is often a trial and error task that is time consuming and hence TPC algorithms become non-generic.

6.2 Overview of DA-TPC

To overcome the disadvantages mentioned in the previous section, DA-TPC algorithm is proposed. Unlike other TPC algorithms such as P-TPC, ART that is window based TPC, DA-TPC is a sender-side per-packet based TPC algorithm that establishes a strong link between a pair of nodes making it more robust to changing environmental conditions. Instead of using variation in link quality as a trigger to adjust the T_x level, the priority of the individual data is used as the only metric to make a decision to boost the transmitting power.

As shown in Figure 6.2, application layer tags the data that it receives from the sensors with a specific priority. Based on the tagged information, appropriate T_x level and routing protocol is activated at the lower layers. For instance, if the temperature of the room is classified as high priority data once it reaches an unusual level. If the temperature of the room is within an expected pre-defined level it is classified as a low priority data. In the case of high priority data, DA-TPC boosts the T_x to the maximum level to make sure that the critical (high priority) data reaches the destination with a higher guarantee. The reliability of low priority data is achieved by the selection of T_x during the initialization phase that ensures establishment of strong link between a pair of nodes.

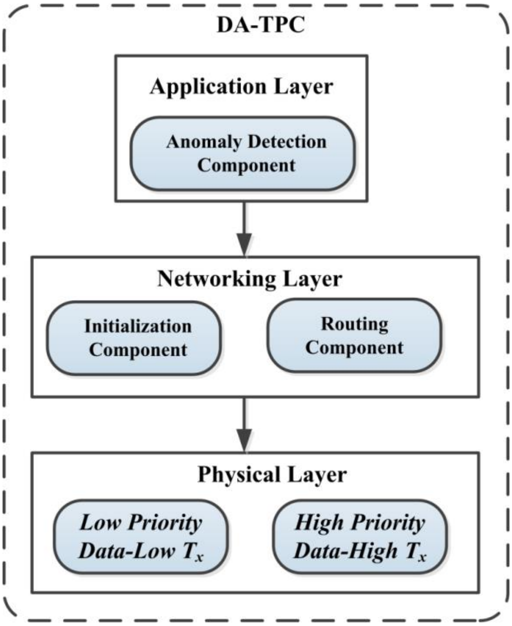


Figure 6.2 Block Diagram of TPC

The design of DA-TPC fulfils following design goals mentioned in chapter 1

a) Minimal communication overhead
b) Minimal effect on other layers such as routing and MAC
c) Capability of instantly and accurately predicting the variation in the sensor data and recommending an appropriate T_x

6.3 Components of DA-TPC

DA-TPC is implemented in TelosB motes with a CC2420 transceiver running Contiki OS. DA-TPC has three main components as shown in Figure 6.3

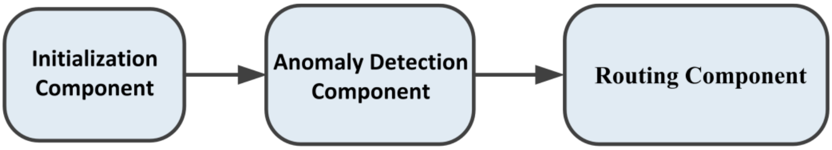


Figure 6.3 Components of DA-TPC

The purpose of the initialization phase is to discover neighboring nodes at the boot up and determine T_x level to use to relay low priority data. After initialization phase, the anomaly detection component at the application layer is responsible for checking the priority of the data. If the priority of the data is low, then T_x selected at the initialization phase is used. Else, to enhance the reliability of the link, maximum T_x available in the transceiver is selected to transmit high priority data. However, increase in T_x does not increase the contention in the network (refer section 6.5). Although appropriate T_x is set at anomaly detection phase, it is executed at the layer 1(physical layer). After tagging the data, routing component is executed to multi-hop the data from the source to destination.

6.3.1 Working of Initialization Component of DA-TPC

The main purpose of initialization phase is to determine a T_x level to be used by the routing protocol to relay low priority data. All nodes in the network perform initialization phase, where each node advertises a probe packet. To learn the variation of link quality, TPC algorithms presented in Table 6.1 transmit a large amount of probe packets at all available power levels. A detailed list of the T_x levels used by the CC2420 transceiver is provided in Table 6.2 [71].

Table 6.1 Partial list of TPC algorithms with prolonged initialization phase

TPC algorithms	Packets sent at initialization phase
ATPC [68]	Approx. 2048 packets per pair of nodes
Distributed TPC [109]	Approx. 1622 packets per pair of nodes
Probe based TPC [41]	Approx. 10242 packets per pair of nodes

Table 6.2 Power consumption and their communication range for CC2420

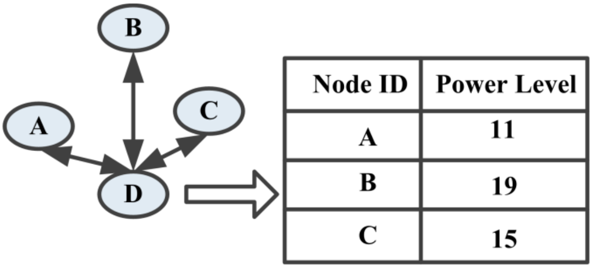
TelosB T_x levels parameter values	3	7	11	15	19	23	27	31
Distance (m)	3	10	16	22	29	34	40	48
Output power (mA)	8.5	9.9	11.2	12.5	13.9	15.2	16.5	17.4

A prolonged initialization phase not only delays the operational time, it also consumes a lot of energy. One of the simplest methods to conserve energy of the nodes is to use only a subset of the T_x levels. A WSN deployment technician would know the approximate distance the nodes are placed from one another. Designing the algorithm based on this fact and knowing the communication range offered by each T_x level of the respective transceiver, one can avoid broadcasting probe packets at every T_x level and thereby consume more energy as it is done in various other TPC algorithms. DA-TPC assumes that the distance between each sensor node is already known and therefore it uses only a subset of T_x levels. Therefore, DA-TPC has a minimal communication overhead. For example, as a proof of concept in this chapter, only three T_x levels (11, 15, and 19) out of 8 available T_x levels provided by the CC2420 transceiver were employed to transmit probe packets during the initialization phase. These packets contain the subset of T_x level values that was used to transmit it.

During boot up, all nodes perform the initialization phase simultaneously by broadcasting three probe packets every three seconds at T_x levels 11, 15, 19 respectively. The neighboring nodes that are within the transmission range of the subset of T_x levels receive unique probe packets from a specific node and store the corresponding T_x levels and their node id.

Consider a small network as shown in Figure 6.4. For instance, let us assume we are interested to find T_x level selected by node D at initialization phase. Node D receives only one unique packet per T_x level from all the neighboring nodes (A, B and C).

Since initialization phase is performed simultaneously by all the nodes, due to the collision, let us assume that only the packet sent at T_x level 11 by node A is received by node D and corresponding packets sent at level 15 and 19 are lost. Similarly, let us assume that the probe packet sent by node B only at T_x level 19 is received by node D. Likewise due to the collision, node C packet sent at level 15 alone is received by the node D.



Node ID	Power Level
A	11
B	19
C	15

Figure 6.4 Working of Initialization phase of DA-TPC

Node D constructs a table in memory that contains the node id and the T_x value that it received from probe packets. DA-TPC employs an initialization phase that does not maintain a comprehensive list of information from all the nodes and hence is more memory efficient. Finally, node D sets the maximum T_x level (19) from the list. The routing protocol in node D then uses this T_x level

(19) to relay a low priority data with any of the nodes based on the shortest path to sink.

In Figure 6.4, imagine node A is 15m apart from source node D that has a low priority data to send. Assume that the routing protocol running in node D chooses node A has the first hop to its destination. As per working of the initialization phase of DA-TPC discussed above, for Node D to communicate with node A, it uses T_x level 19 that has a communication range of 29m (refer to Table 6.2). Not utilizing a minimum power (11) which is just enough to establish a link between a pair of nodes D and An as done by various other TPC algorithms such as MODTPC, AODPTC [128][58] seems counter intuitive. However, by establishing a weak link with minimum transmit power, we run the risk of breaking the communication path due to the collision. This is even more the case when multiple nodes transmit at the same time.

When the link breaks, routing protocol has to transmit extra control messages to find a new path (refer chapter 3). Therefore, maximum T_x level from the subset (11, 15, 19) is chosen to keep the network stable as long as possible. Also as pointed out in chapter 4, low transmission power causes more collision due to hidden-node and one of the simplest methods to root out hidden node is to maximize the transmission range with higher T_x. Similar to DA-TPC, there are TPC algorithms that have shorter initialization phase as shown in Table 6.3.

Table 6.3 Partial list of TPC algorithms with shorter initialization phase

TPC algorithms	Packets sent at initialization phase
ODTPC [57]	2 packets per pair of nodes
AODTPC [58]	2 packets per pair of nodes
BS-TPC [59]	4 packets per pair of nodes
DA-TPC	3 packets per pair of nodes

However, configuring DA-TPC to use a subset of T_x levels based on the network topology is much easier than tweaking the RSSI threshold models that are widely used in existing TPC to adapt T_x. In addition, TPC algorithms in Table 6.3 are receiver based and therefore the receiver node must bare additional transmission cost by responding to source node with appropriate T_x level.

6.3.2 Working of Anomaly Detection Component

Anomaly detection component has two main tasks. First, it checks if the priority of the data is low or high. Second, it is a mapper function as shown in equation 6.1, maps the priority of the data to a specific T_x level and routing protocol to be used by the node to relay the low or high priority sensor data.

$$(P_{Tx}, R) = f(P_r) \tag{6.1}$$

$P_r \in \{P_l, P_h\}$ is the set of priorities that a sensor data is tagged. Here, P_l and P_h represents a low and high priority. The set $P_{Tx} \in \{\mathrm{Init}_{Tx}, \mathrm{Max}_{Tx}\}$ represents the specific T_x levels selected. $Init_{Tx}$ and Max_{Tx} are the subset of power levels determined at initialization phase and the maximum power level available in CC2420 transceiver respectively. The set $R \in \{R_{RPl}\}$ contains Collection Tree Protocol (CTP) routing protocol. By implementing the simple weighted moving average algorithm (WMA) at the sender node, priority of the future data packet is instantly and accurately predicted. Employing WMA is appropriate because of its performance compared to other learning algorithms (refer chapter 5)

6.3.3 Working of Routing Component

Nodes use CTP routing protocol to multi-hop the data from the source node to sink. All the nodes for relaying both low and high priority data, dynamically select the path from the source to destination node. The modular structure of DA-TPC allows usage of any other protocols. However, DA-TPC by default employs CTP.

6.3.4 DA-TPC

Figure 6.5 illustrates the entire workflow of DA-TPC. First, every node executes the initialization phase. Next, the anomaly detection component in the node checks the data priority (low or high) and maps it to a specific T_x level. Lastly, the routing protocol (CTP) forwards the data.

6.4 Experimental and Simulation Setup

DA-TPC algorithm is evaluated for a 10-node network as shown in Figure 6.6. All nodes are separated from another by a distance of 10m. All nodes are configured to transmit one packet randomly in 2 to 4 seconds time interval to sink node 10. To emulate fading of the signal strength with increase in distance between the source and the destination node, Unit Disk Graph Model (UDGM) was used [49]. All nodes in the network have a start-up delay of 1000ms and emulate TelosB sensor motes equipped with CC2420 transceiver [11].

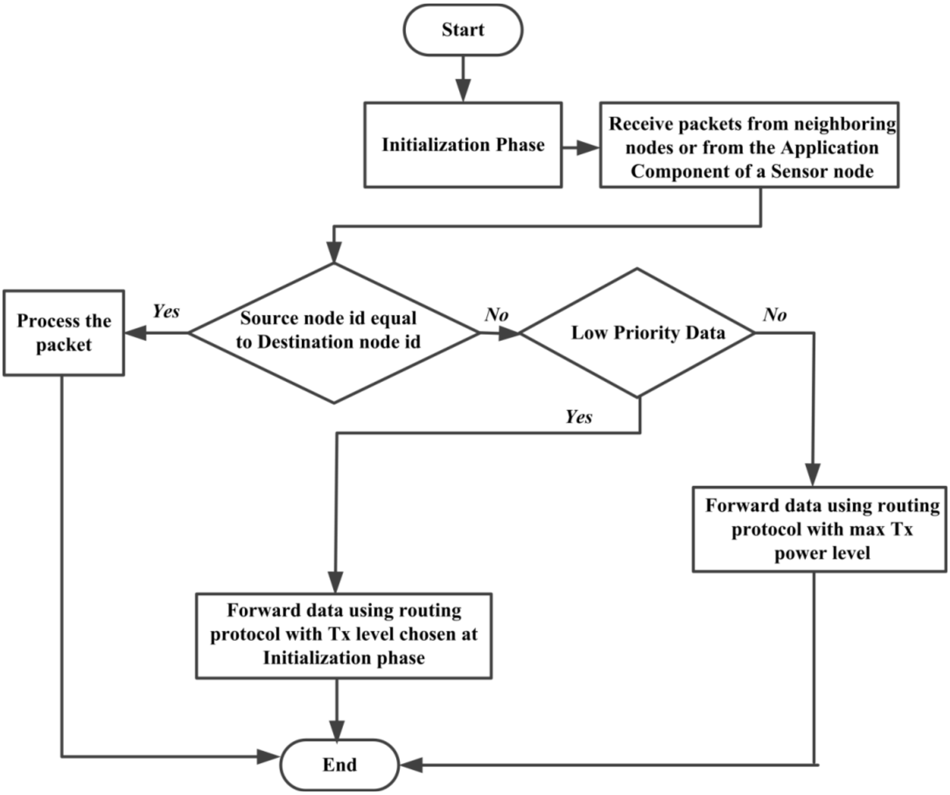

Start
Initialization Phase
Receive packets from neighboring nodes or from the Application Component of a Sensor node
Source node id equal to Destination node id
Yes
Process the packet
No
Low Priority Data
No
Yes
Forward data using routing protocol with max Tx power level
Forward data using routing protocol with Tx level chosen at Initialization phase
End

Figure 6.5 Workflow of DA-TPC algorithm

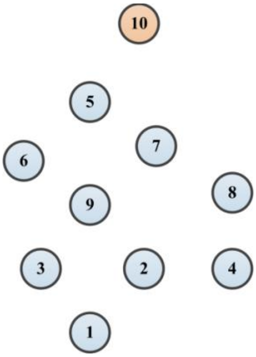


Figure 6.6 10 nodes network setup

DA-TPC was tested with CTP routing protocol and ContikiMAC MAC protocol at lower layers. To check the channel condition, all the nodes perform Clear Channel Assessment (CCA) at the MAC layer [76]. Four different experiments were conducted. At a time only one node, either node 2, 7, 8 and 9 are configured to execute DA-TPC algorithm.

6.5 Results and Discussion

In this section, we compare the performance of DA-TPC at two different levels- network and node level. The comparison is performed in three different scenarios - a network with no nodes adjusting their T_x (No-TPC), a network with only one node (either 2, 7, 8 and 9) adjusting their T_x based on reactive TPC model and DA-TPC model. We also evaluate the energy consumption of initialization phase of DA-TPC with other TPC algorithms with a shorter initialization phase.

Table 6.4 Transmission coverage of the nodes with three different transmission power levels

Transmit power levels	Transmission Coverage (%)			
	Node 2	Node 7	Node 8	Node 9
T_x=15	67	78	45	89
T_x=19	78	89	67	89
T_x=11	On an average 31% for all nodes			

In No-TPC scenario, all nodes use fixed T_x level of 11. Reactive TPC nodes by default employ T_x level 15 and enhance the T_x to 19 when link quality falls. Whereas, in the case of DA-TPC, nodes by default employ T_x level of 19 and adjust T_x to 31 when there is a high priority data to transmit. Table 6.4 provides the transmission coverage at three different transmission powers.

6.5.1 Performance of DA-TPC at Network Level

Figure 6.7(a), Figure 6.7(b) and Figure 6.7(c) presents the average PDR, latency, and duty cycle of the network in three different scenarios- No-TPC, reactive TPC, and DA-TPC model.

PDR of the network (Figure 6.7(a)) is 48% and 37% more in the case of DA-TPC compared to No-TPC and Reactive TPC respectively. Whereas the latency of the network (Figure 6.7(b)) in the case of DA-TPC is less by 29% and 13% compared to No-TPC and Reactive TPC respectively. As far as the duty cycle is concerned (Figure 6.7(c)) the percentage of time the transceiver remains in on-state in the case of DA-TPC is 1% less compared to No-TPC. However, it is 2% more compared to Reactive TPC. The duty cycle is only marginally higher in DA-TPC compared to Reactive TPC for the obvious reason that higher T_x is employed by the nodes to achieve stronger links.

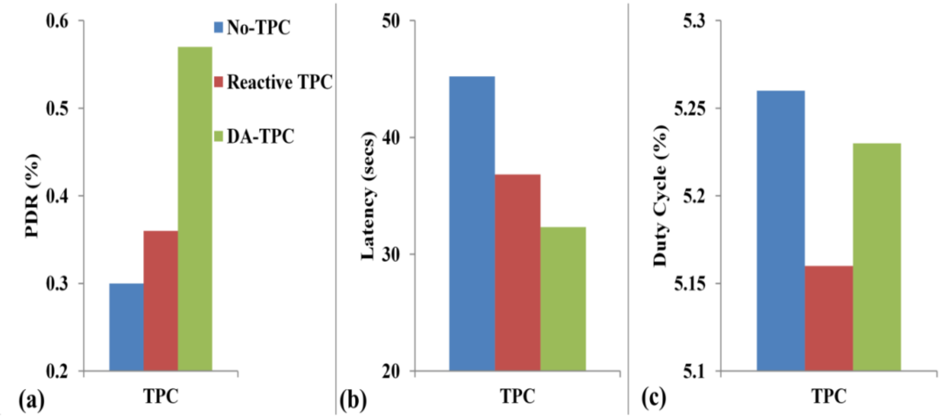


Figure 6.7 Comparison of network wide performance of DA-TPC

The reason for the better performance of DA-TPC is because of the reduction of two crucial problems – hidden-node and avoiding low power links. These two problems are avoided by deliberately adopting much higher T_x level (refer Table 6.4) than what is required to connect to the immediate neighbouring node at the initialization phase. Recollect from chapter 4 that hidden-node problem manifests more strongly in the case where nodes employ lower transmission power ($LowT_x$) than higher transmission power ($HighT_x$). When a node transmits at $HighT_x$, more number of nodes falls in the communication range and they all defer their communication because of Clear Channel Assessment (CCA) enabled at the MAC layer.

To conserve energy, many TPC algorithms proposed in the literature configure the transmitting nodes to utilize $LowT_x$. As pointed out earlier, $LowT_x$ increases the possibility of hidden-node problem and collision is inevitable and hence retransmission increases. This condition is shown in Figure 6.8 that provides information on the total number of retransmitted packets in the network in three different scenarios – network with no nodes performing TPC (No-TPC) and a network with one node (2, 7, 8 and 9) performing TPC (DA-TPC or Reactive TPC). As you can observe, a network with no nodes performing TPC generates the highest number of retransmission compared to the case where one of the nodes 2, 7, 8 and 9 perform TPC either by employing DA-TPC or reactive TPC. However, an exception to this comes when node 7 performs reactive TPC. Recollect from chapter 3 that when a node that relays the traffic of majority of other nodes in the network performs TPC, there is a spike in the retransmission.

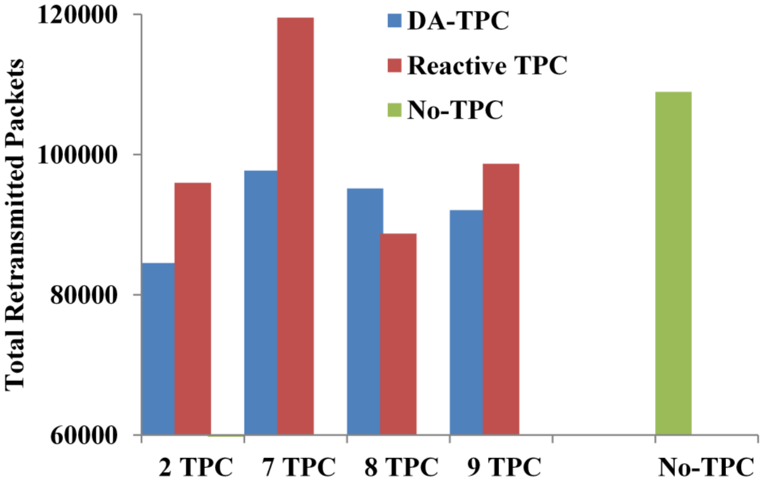


Figure 6.8 Packets retransmission in the network when individual node perform TPC

Compared to reactive TPC, the number of retransmitted packets (sensor data, control packets from routing layer) is less in the network by 11%, 18%, and 6% when the nodes 2, 7 and 9 adjust their T_x based on DA-TPC. However, compared to reactive TPC, retransmission in the network is 6% more in the case of node 8 adjusting its T_x based on DA-TPC.

6.5.2 Performance of TPC at Node Level

In this section, analysis of one node performing TPC to enhance its own link quality is analysed. Figure 6.9(a), Figure 6.9(b) and Figure 6.9(c) presents the PDR, latency and energy consumption of each individual nodes (2, 7, 8 and 9) respectively when they do not adjust their T_x level (No-TPC) or when they adjust T_x level either using Reactive TPC or DA-TPC.

From Figure 6.9, all individual nodes that use DA-TPC to adjust their T_x level achieve superior PDR with lower latency and energy consumption. When node 2 performs DA-TPC, the PDR is 80% more than what node 2 would have achieved with Reactive TPC. Similarly, when node 2 does not perform dynamic power adjustment, its PDR is 60% lesser compared to the case when node 2 performs DA-TPC.

However, the performance of dynamic power adjustment performed by node 2 either by Reactive TPC or DA-TPC method on an average is 94% and 96% less compared to all other individual nodes (7, 8 and 9) performing TPC either by Reactive TPC or DA-TPC method respectively. This is because in a relatively high contention network the nodes those are farther away from the sink node suffer the most due to the collision.

Latency presented in Figure 6.9(b) is calculated as total packets received at the sink node divided by total packets sent. Although the number of packets received by sink node 10 from nodes 2, 7, 8 and 9 performing DA-TPC is more, the latency of only node 7, 8 and 9 is less compared to their counterparts. This is because node 2 is 10m farther away from nodes 7, 8 and 9 and hence the data of node 2 has to multi-hop more than other nodes.

As far as the duty cycle is concerned in Figure 6.9(c), No-TPC and Reactive TPC employ T_x level 11 and 15 respectively. This results in fewer numbers of nodes in transmission range causing collision due to hidden-node. As collision increases retransmission occurs and hence higher energy is consumed. However, this is not case with DA-TPC as T_x level 19 increases the transmission range and results in the elimination of hidden-node and thereby reduces energy expensive retransmission.

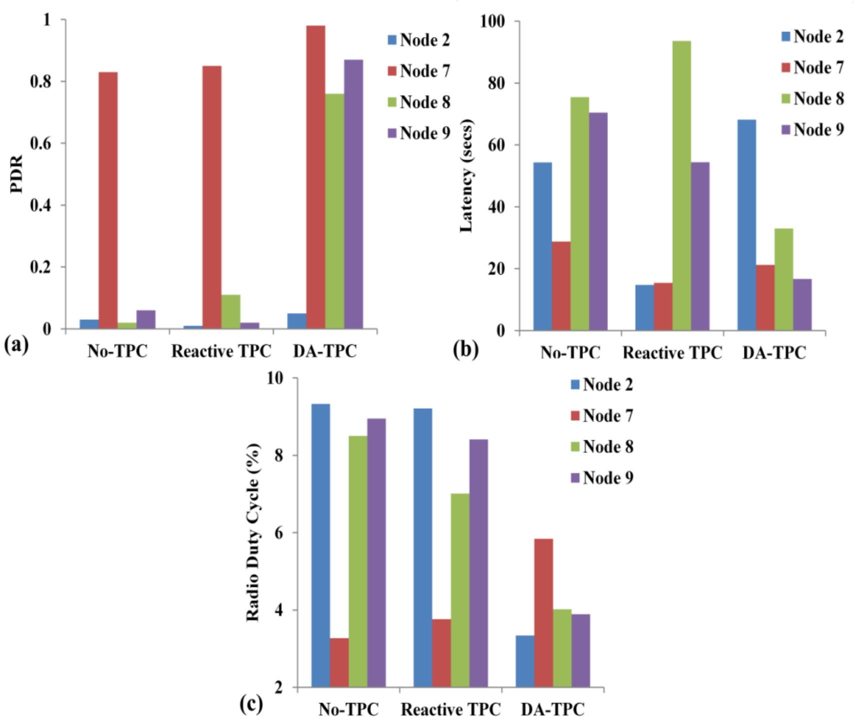


Figure 6.9 Node performance metrics (a) End-to-end PDR (b) End-to-end latency (c) Radio duty cycle

6.5.3 Energy Consumption at Initialization Phase

Finally, we compare the energy drawn during the initialization phase between a pair of nodes having CC2420 transceiver using equation 6.2

$$\sum_{i=1}^{p} TP_{sn} + TP_{rn} \tag{6.2}$$

Here TP_{sn}, TP_{rn} represent energy spent during transmission of probe packets by the source and the receiver nodes respectively. P is the total number of probe packets sent. Energy consumption at initialization phase of DA-TPC along with other TPC algorithms with shorter initialization phase presented in Table 6.3 is shown in Figure 6.10.

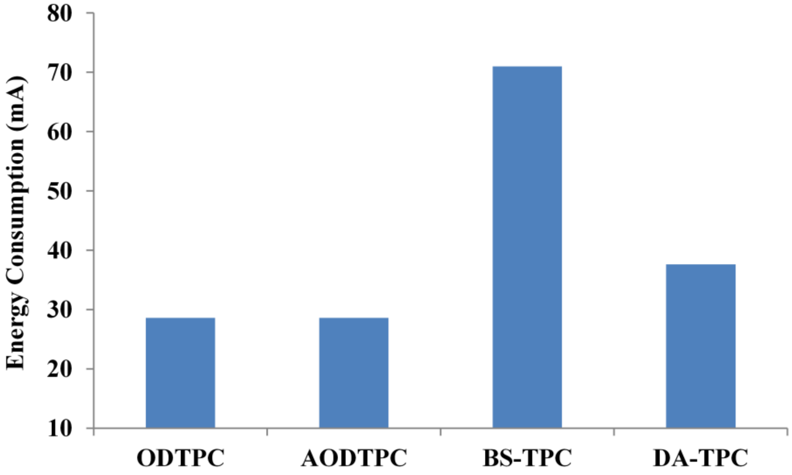


Figure 6.10 Energy consumption of TPC algorithms at initialization phase

From Figure 6.10, we find that DA-TPC consumes 48% less energy compared to BS-TPC at initialization phase and DA-TPC consumes 24% more energy compared to ODTPC and AODTPC. However, ODTPC and AODTPC employ sensitive RSSI metric to determine the T_x to be adopted by a pair of nodes. As sensitive parameters vary abruptly, TPC algorithms adjust T_x frequently.

Chapter 3 shows that frequent adjustment of transmission power can impede the performance of the network. In addition, initialization phase of ODTPC, AODTPC is designed to achieve just enough T_x level to communicate with its neighbours. These low power links are prone to failure due to internal interference. DA-TPC eliminates the low power links and purposefully employs higher T_x to achieve stable links at the cost of higher energy consumption.

Adoption of higher transmission power also eliminates the hidden-node problem.

6.6 Conclusions and Future Work

In this chapter, a new TPC algorithm is dubbed DA-TPC is proposed. DA-TPC uses priority of the data as a metric to adapt the transmission power instead of relying on an environmentally sensitive metric such as LQI, RSSI. TPC algorithms in the literature employ just enough transmit power level to communicate with the neighbours. However, this design decision fails in a network with multiple source nodes simultaneously transmitting the data because lower transmission power level often gives rise to the hidden-node problem causing collision and retransmission. DA-TPC reduces the hidden-node issue with the adoption of higher transmission power level. One node in the network performing frequent adjustment of transmitting power level by reactive TPC method impedes the performance of the network. However, as DA-TPC is a sender-side per packet power control algorithm it has a very marginal impact on the overall performance of the network. Unlike other TPC algorithms proposed in the literature, DA-TPC has a shorter initialization phase and therefore as minimum communication overhead.

However, for the Initialization phase of DA-TPC to work accurately the approximate distance between neighbouring nodes must be known in advance. Therefore, DA-TPC will fail to perform in a network with randomly distributed nodes. DA-TPC algorithm was tested for a small-sized static network with CTP and ContikiMAC protocols at lower layers. The performance of DA-TPC in a larger network with other routing protocols such as AODV, RPL (IPv6 *Routing Protocol* for Low-Power and Lossy Networks) and MAC protocols such as B-MAC must be experimented with to evaluate the robustness of DA-TPC.

Chapter 7 Predicting the Energy Depletion of IoT Devices

In this chapter, we investigate a generic method of predicting battery depletion rate of neighboring sensor nodes. To this end, the possibilities of utilizing Received Signal Strength Indicators (RSSI) as an input to a variant of machine learning algorithms known as classification algorithms to infer the state of the battery of embedded devices is evaluated. Smoothed RSSI is used as input to these algorithms.

The contribution of this work lies in determining if RSSI values can be used as a potential parameter to represent the energy depletion rate. In addition, comparative analysis of various well-established classification and regression algorithms such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN) for predicting the voltage level of a remote node using exclusively RSSI values is analyzed. These algorithms are evaluated based on their accuracy and speed. As the classification algorithms are computationally intensive, the tasks of estimating the critical state of the battery of individual nodes are diverted to more powerful nodes such as sink node. To this end, this chapter addresses the following research question mentioned chapter 1

(RO6) *How accurately can machine-learning algorithms forecast the battery level of the IoT device?*

The experiments conducted shows that the nature of RSSI values obtained at various battery levels (3V to 1.5V) of the node cannot aid the machine learning algorithms to classify the exact energy level. Therefore, the original assumption that RSSI values can be used as a potential input to machine learning algorithms to predict the gradual energy depletion rate is not valid. However, the algorithms are able to classify the drastic fall (from 3V to directly 1.5V). Amongst the nine learning algorithms, speed and the classification accuracy of determining the drastic fall by SVM and logistic regression are the highest, whereas linear regression has the worst performance compared to all other algorithms evaluated in the experiment.

Remaining sections of this chapter are as follows. In section 7.1 importance of predicting the energy level is outlined. Background information on other popular battery power level prediction techniques is explained in section 7.2. Section 7.3 provides an architecture of sink node based battery level estimation technique. The experimental setup is presented in section 7.4. A brief overview of various machine-learning algorithms that are used in the experimentation is provided in section 7.5. Prediction accuracy of various machine learning is discussed in section 7.6. In section 7.7 we conclude and provide further research ideas on this topic.

7.1 Importance of Predicting Energy Level

Imagine a large scale IoT network deployed to monitor temperature and humidity, with an intention to optimize the yield of farming land such as potato field or a vineyard [22][87]. When IoT devices such as sensor nodes operate in outdoor scenarios, there are many points of failures. One of the main causes of failures could be an unpredictable battery depletion of individual devices.

Many of the IoT devices such as TelosB operate with a limited battery capacity. In a dense network, the possibility of collision is higher and therefore the possibility of retransmission is higher and every additional transmission incurs energy. For a device such as TelosB mote, in default factory setting the transmission consumes 18.4 mA, higher than any other components of a mote [11]. In addition, to thwart security breaches, individual devices run computationally expensive encryption and authentication programs that consume a considerable amount of energy [129][10]. Furthermore, variants of Denial-of-Sleep malicious attack that keeps the IoT devices’ radio in unnecessarily prolonged wake-up/listen state can drain the battery in only in a couple of days [10]. These concerns can easily deplete the battery sooner than expected. Replacing the depleted battery of the nodes that are deployed in remote and harsh environmental conditions is not always feasible, due to the operational cost.

Failure of strategic nodes due to above-mentioned concerns can bring down the entire network and is not favorable for the end users who depend on it for day-to-day operation. Figure 7.1 depicts a small sized network and illustrates the impact of the failure of a strategic node. As you can see from Figure 7.1, node 2 has a high degree of centrality as it relays the traffic of the majority of nodes. Battery depletion of this node would severe the network if other neighboring nodes cannot find any alternative paths, for instance, due to limited transmission range or routing issues.

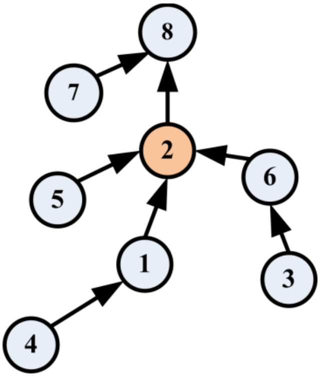


Figure 7.1 Traffic flow in a small size network

Early prediction of the battery depletion rate can be advantageous for following reasons

1. Rerouting the traffic through more energy efficient paths, before the network is disconnected [130].
2. Assisting the TPC algorithms to make a more informed decision on adjusting the T_x level based not only on the channel condition but also the physical condition of the device.

Therefore, the concerns and potential benefits uphold the need of predicting energy depletion.

7.2 Literature Overview

Figure 7.2 depict various techniques pointed out in the literature to predict the battery life of a device. The first well-known technique is battery modelling. A mathematical model of a battery discharge rate is constructed that can derive the remaining life time of the battery [131].These mathematical models capture the discharge rate broadly in two ways - offline and online.

An analytical battery model that can accurately estimate the battery discharge behavior is proposed in [132][133]. However, the major demerit of this model is that it is offline, as it requires long computation time. Therefore, offline models have limited utility for the implementation in low power wireless networks. To overcome the obvious disadvantage of offline analytical models, a wide array of online computation of discharge rate are proposed [130][134][135].

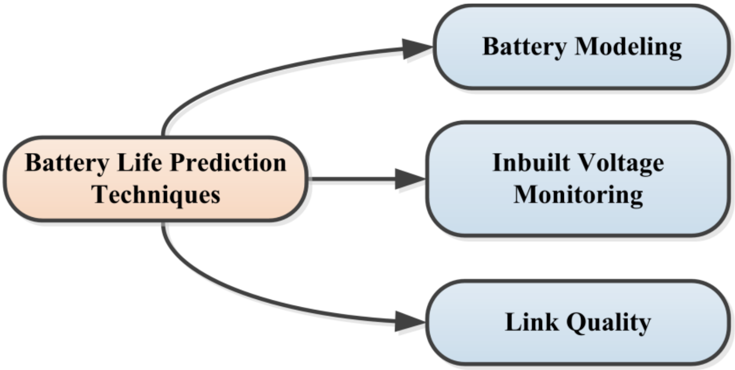

Battery Life Prediction Techniques
Battery Modeling
Inbuilt Voltage Monitoring
Link Quality

Figure 7.2 Various battery-monitoring techniques

Both offline and online battery energy models provide decent prediction accuracy only in simulators, as they do not imply variable environmental conditions such as temperature within cold logistic units. The battery of sensor nodes is subjected to extreme conditions when they are deployed to monitor shelf life of perishable goods in cold storage units, monitoring of climatic condition in the dry and hot area. It is a well-known fact that temperature greatly affects the battery behavior [134][21]. Constructing a model that includes all the parameters that reflect not all environmental conditions are feasible due to acute resource constrains of the device. As there are various battery models (Nickel-Cadmium (NiCd), Nickel-Metal-Hydride (NiMH), Lithium-Ion (Li-Ion)), these energy models are not generic and require adaptation to suit a specific battery model.

The second technique to forecast remaining battery life is to activate the internal voltage-monitoring sensor. Devices such as TelosB have MSP430 microcontroller that has an internal voltage monitoring sensor- supply voltage supervisor (SVS) circuitry that can detect if the supply voltage drops below a predefined user selectable level. However, SVS circuit is specific to MSP430 microcontroller making it a non-generic solution. Enabling battery-monitoring sensor on resource constraint nodes is not advisable as they consume extra energy. The method presented in this chapter offloads the task of monitoring the battery to more energy rich nodes- a sink node.

The last technique to forecast the remaining battery life is to utilize RSSI to determine the battery depletion rate. This metric is provided by most of the transceivers of wireless devices such as TelosB motes. They are embedded in all the incoming packets and can be extracted any time. It represents the strength of the incoming signal and is measured in dBm. As RSSI is strongly affected by the battery discharge, it can be used to monitor the depletion rate of a neighboring node. These neighboring nodes can then exchange their energy detail with one another for various purpose such as routing [136]. The relationship between transmit power and supply voltage is discussed in [137]. Ease of calculating and utilizing RSSI to forecast the energy depletion makes it a viable solution, as it does not require employing a sensor that monitor the voltage or constructing new models.

To that extent, various machine learning algorithms such as linear regression were used to predict the depletion rate [138]. However, it is known that RSSI are sensitive to environmental conditions such as temperature [33]. As RSSI can vary due to multiple factors, it is challenging to validate if the variation in RSSI was indeed due to depletion of the power source. It is necessary to re-evaluate the usability of RSSI for predicting remaining battery life as this not yet done. Therefore, evaluating the accuracy of this method is the main goal of this chapter.

7.3 System Overview

Consider a partial view of randomly deployed large IoT network as shown in Figure 7.3. Each of the numbered nodes is responsible for monitoring environmental data and transmitting it to the cluster nodes (1 and 6). The aggregated data from the cluster nodes are sent to the sink node for further processing.

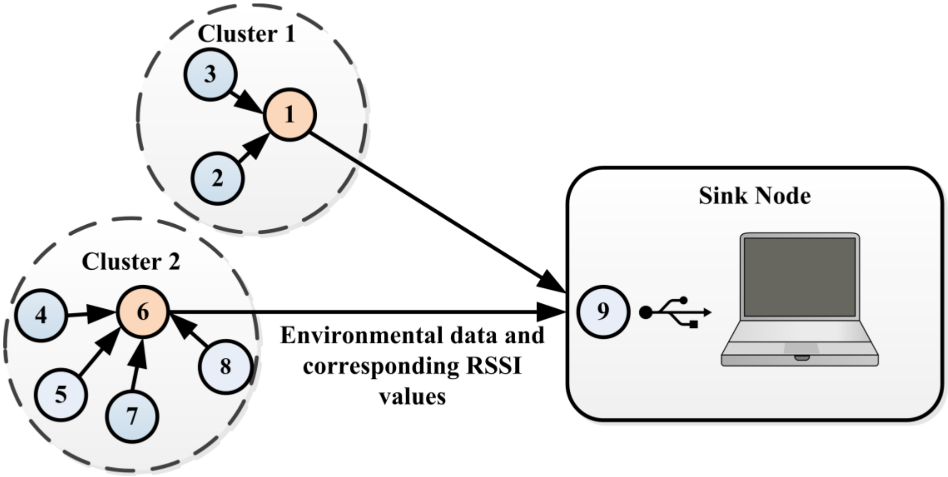


Figure 7.3 Envisioned architecture of a battery depletion prediction system

As pointed out earlier, RSSI values are non-linear and can vary abruptly due to multiple factors such as temperature, humidity [21][22]. Providing the machine learning algorithms with sharp variations can give a less accurate prediction. Therefore, to reduce the number of outliers and enhance the prediction accuracy of the machine learning algorithms, the data must be smoothed. Due to sharp variations in the RSSI values, it is not possible to classify whether the battery level is good, average, or bad with single RSSI value. Hence, we need to maintain a window that keeps the most recently read RSSI values. To meet the above-mentioned requirements, the algorithm of choice for our experiments is Adaptive Window (ADWIN) as it uses the concept of a sliding window. Unlike simple moving average that has a fixed window size *w* to calculate the average, ADWIN uses variable window. The window size is recomputed online depending on the rate of change detected from the data in the window itself. This way ADWIN helps in identifying distribution change when learning from the sequence of data [139].

The sequence diagram of the system is shown in Figure 7.4. Sensor nodes monitor the environmental condition and transmit it to the cluster node.

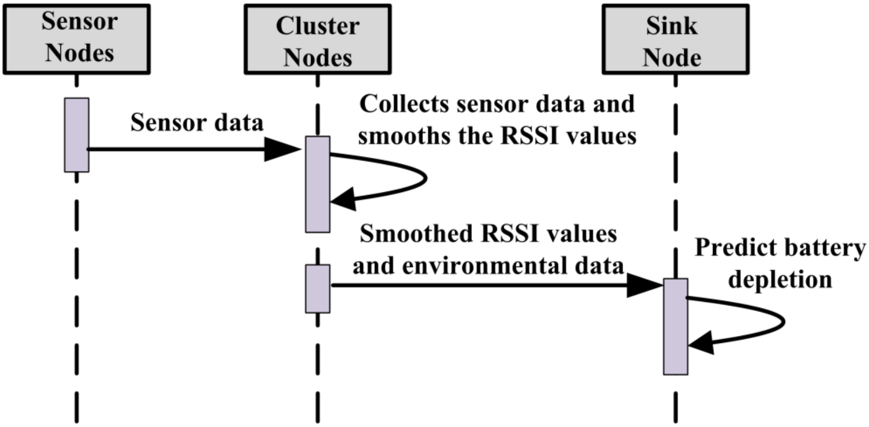

Sensor Nodes
Cluster Nodes
Sink Node
Sensor data
Collects sensor data and smooths the RSSI values
Smoothed RSSI values and environmental data
Predict battery depletion

Figure 7.4 Sequence diagram of envisioned energy depletion prediction system

A normal sensor node is elected to be a cluster node depending on its remaining battery level and other parameters. For example, a Low Energy Adaptive Clustering Hierarchy (LEACH) protocol can be used to elect a particular node to be a cluster node [140]. Along with the LEACH protocol, the ADWIN algorithm resides in the cluster nodes and smooths the RSSI values of the respective nodes, and transmits these to the sink node. Since machine-learning algorithms are computationally intensive processes, they are executed on a relatively more resource rich devices such as a sink node. Sensor nodes attached to the laptop via USB interface is a sink node (refer Figure 7.3). The data received from the mote can be transferred to the machine-learning algorithm in real-time via USB interface.

7.4 Experimental Setup

RSSI is the sum of the pure received signal strength and the noise floor. The noise in the sensor network communication is introduced due to co-location of other 802.11 network Bluetooth devices and domestic appliances such as microwave oven [16][141]. In addition, concurrent transmission from other nodes within the network can also introduce noise into the communication channel [142]. Prior to assessing the applicability of RSSI values in predicting the battery depletion rate in a realistic network, we need to evaluate how the RSSI values fluctuate in an ideal condition-interference free environment. Therefore, for the sake of simplicity, only the communication between one cluster node (transmitting) and the sink node (receiving) was performed in an anechoic chamber. Hence, Figure 7.5 shows the actual experimental setup. Once the RSSI values were collected, smoothing of RSSI values and prediction of battery depletion was carried offline.

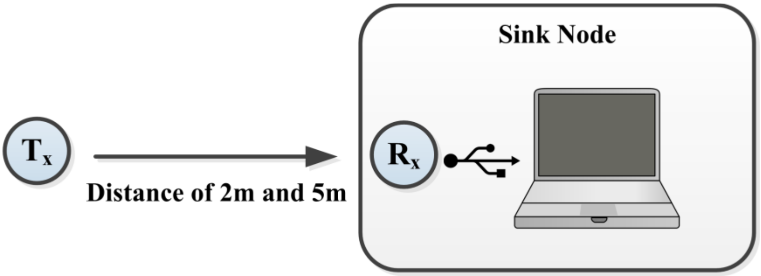


Figure 7.5 Experimental setup in an anechoic chamber

Two sets of experiments were conducted using TelosB motes (T_x and R_x) equipped with a CC2420 transceiver. In the first setup, the distance between transmitting node (T_x) and receiving node (R_x) was set to 2 meters. In the second setup, the distance was increased by 5 meters. The battery depletion rate of the transmitting node was emulated by using Benchmark power supply. The following settings were kept constant for every set of experiment

1. Transmitting node was configured to send data to the receiving node every 250ms at maximum transmit power.
2. The receiving node connected to the laptop was our sink node (refer Figure 7.5).
3. Once the experiment is started, the position of the transmitting and the receiving nodes was not changed during the entire duration of the experiment.
4. The current of TelosB motes was set at 0.25mA.

5. For every voltage ranging from 3V to 1.5V (max and min operational voltages of TelosB mote), and two distances (2m and 5m), 1000 RSSI reading were taken.

7.5 Overview of Machine-Learning Algorithms

The machine learning algorithms used in this experimentation can be broadly classified into two groups –classification (SVM, GMM, RFT, KNN, and RC) and regression (LR and LogR). Please see Table 7.1 for the acronyms description. Sci-Kit Python machine learning framework was used for predicting the energy level [143].

Table 7.1 Legends for Table 7.2, 7.3, 7.4, and 7.5

SVM-RBK: Support Vector Machine with Radial Basis function
SVM-PK: Support Vector Machine with Polynomial function
GMM: Gaussian Mixture Model
RFT: Random Forest Trees
KNN: K Nearest Neighbors
LogR+RBM: Logistic Regression built on a top of a Restricted Boltzmann Machine
LogR: Logistic Regression
LR: Linear Regression
RC: Random Classifier
Not available (NA): we stopped the algorithm due to the running time bigger than 1 minute

The classification algorithms predict the most probable category, class or label for a new data item and the regression algorithms predict the value of a future data item [144].

The first classification algorithm we use is SVM [145]. SVM produces an input-output mapping function from a set of labelled training data. The mapping functions chosen for this experiment are polynomial and radial basis function. This is because they are helpful in classifying non-linear data points such as RSSI [144]. For a simple classification task with only two features, the hyperplane can be a function as shown in equation 7.1 that represents a straight line. In order to classify the data into two classes, SVM learns the hyperplane.

$$y = mx + b \tag{7.1}$$

The second classification algorithm is GMM. It is a basic classification algorithm that can be used to classify a wide array of N-dimensional data points. Expectation-Maximization function is used for fitting a variety of Gaussian models [146]. GMM is defined as per equation 7.2

$$p(\mathrm{x}) = \sum_{k=1}^{k} \pi_k N\left(\mathrm{X} \mid \mu_k, \Sigma_k\right) \tag{7.2}$$

In equation 7.2, $\pi_1,...,\pi_k$ are the mixture coefficients and each Gaussian density $N\left(\mathrm{X} \mid \mu_k, \Sigma_k\right)$ having mean μ_k and covariance Σ_k .

The third classification algorithm we use is RFT and it is an ensemble of decision trees, which will output a prediction value. Each decision tree is constructed by using a random subset of the training data. The objective of combining decision tree is to increase the prediction accuracy [147].

The fourth classification algorithm is KNN and it is a non-parametric algorithm that stores all the available cases and classifies new case based on the similarity measures such as Euclidean distance function as shown in the equation 7.3 [148].

$$\sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2} \tag{7.3}$$

In equation 7.3 x_1, x_2 and y_1, y_2 represent the coordinates of two dimensional data points.

The fifth algorithm we experiment with is LR. The objective of LR is to explain the relation between one dependent variable and one or more independent variables. Regression analysis is the task of fitting a single line through a scatter plot. LR assumes that a linear relationship exists between the dependent variable and independent variable [149]. A simple linear regression is shown in equation 7.4.

$$y = \alpha + \beta x \tag{7.4}$$

In equation 7.4, y is a dependent and x is an independent variable. α is the intercept i.e., the value of y when x=0. β is the slope of the line and it represents the rate of increase or decrease in the value of y for each unit increase in x.

Finally, we use LogR. Like LR, LogR is also used for regression. LR assumes that the dependent variable is normally distributed. However, in many situations dependent variable is not normally distributed and can represent probability value in the range {0, 1}. LR assumes that a constant change in the independent variable will also result in the constant change in the value of a dependent variable. LogR eliminates this assumption by relating a linear combination of the independent variable to the dependent variable by a link function such as RBM [145]. A simple logistic regression is shown in equation 7.5.

$$F(t) = \frac{1}{1 + e^{-\sum_{j=0}^{M} \beta_j x_{ij}}} \tag{7.5}$$

Here, e is the natural log and β is the regression coefficient. $i = 1 \ldots N$ is the number of observations and $j = 1 \ldots M$ is the number of individual variables. Finally, we test the prediction accuracy of all the algorithms with the prediction accuracy of a random classifier. As the name suggests, the random classifiers randomly classify the data points into one of the known classes. The random classifier is used as a benchmark to test the speed and accuracy of all other algorithms [145].

7.6 Evaluation

This section presents the experimental results from conducting the aforementioned experiments. The section is split into two parts: data smoothing and battery level prediction. Data smoothing is executed at the cluster node level and aims at reducing either the processing or the communication or both. Moreover, it contributes to the efficiency of the prediction algorithm by reducing the outlier data points and, hence, the overlap of the classes used at classifiers or reducing the bias in the regression models. Smoothened data are the input to the prediction algorithms that are running in the sink node. The algorithms are evaluated based on their accuracy and speed.

Figure 7.6a presents the raw input RSSI data in the two datasets as well as the smoothing of that data using a naive moving average (Figure 7.6b) and the ADWIN (Figure 7.6c and Figure 7.6d) algorithms. The moving average algorithm outputs the average value of a window of the 10 latest samples for every new RSSI value received. The ADWIN algorithm is used in two modes:

-*Verbose:* for every RSSI value received, ADWIN outputs the average value of the current window (Figure 7.6c).

-Change-detection: ADWIN provides the average value of the last window just on moments of a change at the window size (Figure 7.6d).

As shown in Figure 7.6a, the two RSSI raw datasets are overlapping. On the one hand, the moving average algorithm filters out many outliers that were causing that overlap. On the other hand, ADWIN has reduced significantly the variance of the two datasets and has increased the gap in between.

As expected, the data-points generated by ADWIN in Figure 7.6d are significantly fewer than those in Figure 7.6c as data are submitted to the prediction algorithms in the sink node solely upon a considerable change to the ADWIN window size.

An ADWIN window changes upon a shift of the estimated voltage level, i.e. concept, based on the received RSSI values. Had such concept shift not been present, there would also be no need for triggering the battery voltage level prediction algorithm. Therefore, ADWIN on change-detection mode reduces the communication overhead for the sensor nodes and the processing overhead for the prediction algorithms.

The input data shown in Figure 7.6 are the training data for the prediction algorithms. Every training data-point in those datasets is classified to one of the 16 voltage levels (1.5v-3v). Therefore, any RSSI value from the testing datasets has to be fed into the prediction algorithm and classified to one of those levels i.e. classes. The output of ADWIN algorithm in both modes was used for the classification process.

Table 7.2 and Table 7.3 (please see Table 7.1 for the acronyms description) present the evaluation of various algorithms concerning their accuracy (percentage of input data-points classified in the correct class) and execution time (seconds spent during training phase). From Table 7.2 we find that when smoothed (ADWIN in verbose mode) RSSI values is provided to classifiers to classify the RSSI values to one of the 16 voltages levels (1.5V to 3V), the prediction accuracy of all the nine algorithms are not impressive. This is because the overlapping of data points (RSSI) even after smoothing the data is high for any learning algorithms to accurately predict.

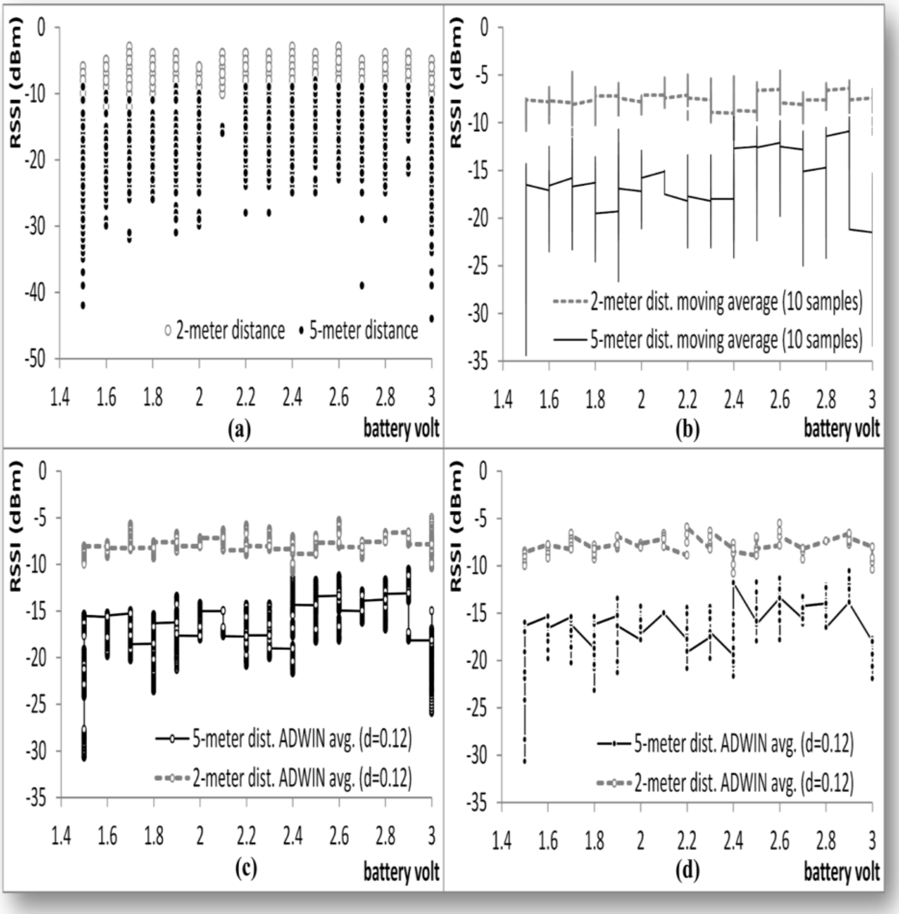


Figure 7.6 Raw and pre-processed input data from the two monitored network conditions. Figure 7.6a illustrates all raw data points. Figure 7.6b presents the moving average of those datasets with a sliding window of 10 samples. Figure7.6c depicts the output of ADWI

On the other hand, Table 7.3 demonstrates a slightly improved situation. The prediction accuracy of the classifiers is better when smoothed RSSI values obtained from the ADWIN algorithm in change detection mode is provided as input to all the classifiers. This is evident only when the distance between the nodes is five meters as overlapping is much reduced (refer Figure 7.6d). However, even in that case (ADWIN in change-detection mode), classifier's performance is limited.

Table 7.2 Evaluation of prediction algorithms. Input data come from the output of ADWIN in verbose mode. RSSI values are classified to one of the 16 voltage levels i.e. classes

Classification Algorithms	2-meter distance dataset		5-meter distance dataset	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	10.76	3.7464	10.51	3.7499
SVM-PK	NA	>1 minute	NA	>1 minute
GMM	5.51	2.3817	7.27	3.2728
RFT	10.23	0.3366	9.76	0.3472
KNN	10.19	0.0108	9.89	0.0107
LogR+RBM	12.86	4.1106	13.99	4.0429
LogR	12.71	0.1648	12.36	0.1727
LR	5.62	0.0277	10.56	0.0221
RC	6.25	>1 minute	6.25	>1 minute

Table 7.3 Evaluation of prediction algorithms. Input data come from the output of ADWIN in change detection mode. RSSI values are classified to one of the 16 voltage levels i.e. classes

Classification Algorithms	2-meter distance dataset		5-meter distance dataset	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	7.86	0.0010	12.10	0.00309
SVM-PK	17.97	0.4803	NA	>1 minute
GMM	8.98	0.0523	8.28	0.0630
RFT	17.97	0.0050	15.92	0.0050
KNN	12.35	0.0007	12.10	0.0005
LogR+RBM	12.35	0.0492	14.01	0.0799
LogR	11.23	0.0492	13.37	0.0034
LR	7.86	0.0019	12.74	0.0003
RC	6.25	>1 minute	6.25	>1 minute

The default communication channel 26 of 802.15.4 was used by the nodes (transmitter and the receiver) to send and receive the packets. Channel 26 does not interfere with European WLAN 802.11 network [150]. However, there are still various reasons behind this inaccuracy. The input RSSI values have a very high variance for each voltage levels [75]. This variance, in a well-controlled environment like the anechoic chamber, might be caused by the inaccuracy of

RSSI register at the receiver, which, in TelosB nodes, varies for plus or minus 6dBm. In addition, it is shown that 802.15.4 compliant radios such as Atmel AT86RF230 [74], Chipcon CC2420 [71] introduce systematic errors in their RSSI measurements [72]. Moreover, the average RSSI value of any voltage level differs maximum 3dBm from any other level. These two issues create a very wide overlapping among the voltage classes that not all the tested classifiers can easily detect.

Therefore, the results in Table 7.2 and Table 7.3 are inconclusive concerning the inference of the battery level of a neighboring sensor node using only received RSSI values at an early stage. However, during the experiments, we noticed that two voltage levels were more accurately predicted than others were. As shown in Table 7.4 and Table 7.5, the classifiers can perform much better, when just two classes are considered. Instead of 16 classes, the classifiers were trained with the same input data to classify data-points into either the 1.5V-1.6V class or the 1.7V-3.0V class. That classification can practically predict if the battery of the remote sensor node has maximum 0.2V before it is drained. However, this is not of a much use because the aim is to accurately determine early in time the energy depletion before the sensor nodes reach their cut-off point so that some preventive action be taken.

Table 7.4 Evaluation prediction algorithms. Input data come from the output of ADWIN in verbose mode RSSI values are classified to one of two classes (1.5-1.6V or 1.7-3V)

Classification Algorithms	2-meter distance dataset		5-meter distance dataset	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	89.11	1.5670	86.08	2.1382
SVM-PK	NA	> 1 minute	NA	> 1 minute
GMM	67.08	0.4543	64.14	0.4907
RFT	83.19	0.1664	79.19	0.2405
KNN	81.10	0.0109	76.95	0.0110
LogR+RBM	87.44	2.8912	87.41	4.8276
LogR	92.44	0.1741	84.65	0.1659
LR	24.75	0.0218	26.03	0.0220
RC	50.00	> 1 minute	50.00	> 1 minute

Table 7.4 present an accuracy of tested classifiers up to 92.4% for the 2-meter distance dataset and up to 87.4% for the 5-meter distance dataset. The benefit of using ADWIN in change-detection mode is shown in Table 7.5 as the accuracy

or execution time of many algorithms is considerably improved compared to Table 7.4. This suggests that the energy depletion level can be predicted when there are fewer classification classes. This also means that the gradual decrease in the energy level is often harder to predict.

Table 7.5 Evaluation prediction algorithms. Input data come from the output of ADWIN in change detection mode RSSI values are classified to one of two classes (1.5-1.6V or 1.7-3V)

Classification Algorithms	2-meter distance dataset		5-meter distance dataset	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	86.51	0.0004	85.35	0.0009
SVM-PK	92.13	0.3010	78.34	10.8300
GMM	68.53	0.0137	73.88	0.0206
RFT	88.76	0.0040	85.35	0.0040
KNN	88.76	0.0006	84.71	0.0005
LogR+RBM	91.01	0.0387	85.98	0.0629
LogR	89.88	0.0019	85.98	0.0032
LR	29.21	0.0003	31.84	0.0003
RC	50.00	> 1 minute	50.00	> 1 minute

7.7 Conclusions and Future Work

In this chapter, the prospectus of employing RSSI values to predict the depletion of the battery is presented. Through this experimentation, it can be concluded that the nature of the RSSI values obtained from the transmitting node when its battery level is between 3V and 1.8V cannot help the prediction algorithm to predict node's current voltage level. On the contrary, the RSSI values obtained from the node when its power level reach the cut-off point (1.6V and 1.5V), can help the majority of the classification algorithms detect the sudden fall in the battery level. However, the predicting the battery level at 1.6V does not serve any purpose. This is because minimum operational voltage required by TelosB sensor nodes must be 1.6V. Therefore, the hypothesis that employing RSSI as an input parameter to learning algorithms for early prediction of battery level for various purposes (refer section 7.1) fails.

In addition, it was found that it is not possible to classify whether the battery level is good, average, or bad with a single RSSI value. Therefore, we needed to maintain a window that buffers the most recent RSSI values.

In the course of the experiment, it was discovered that providing the classification algorithm with raw RSSI values reduces the prediction accuracy of the algorithms. The reason for this is large number of outliers. To prune the outliers and to significantly reduce the overlapping data points (RSSI values) for training algorithms, we used ADWIN algorithm because of its unique capability of detecting concept drift in data points.

When the algorithms had to predict from only two classes (1.5V and 3V) instead of sixteen classes(1.5V to 3V), the prediction accuracy of variants of SVM and Logistic Regression is the highest, followed by Random Forest Tree, K-Nearest Neighbor, and Gaussian Mixture Model. Linear Regression has the least accuracy. As SVM and Logistic Regression use non-linear kernel or functions such as Radial Basis, Polynomial, and the Restricted Boltzmann Machine, it can capture the non-linearity of the RSSI values well. The linear regression assumes that the relation between a dependent and the independent variable is linear. However, as the variation in RSSI is non-linear, the assumption of linear regression causes the inferiority in the prediction accuracy.

The experiments were carried out in an anechoic chamber. The RSSI values obtained in the interference free room even after smoothing it cannot aid the learning algorithms to successfully classify the energy level. The prediction accuracy of the algorithms in the real world wireless deployment will be inferior to the results obtained in the anechoic chamber due to inherent nature of RSSI values. Hence, employing it as an input to learning algorithms is not recommended. Constructing an energy model based on battery or using Supply Voltage Supplier (SVS) circuitry makes the solution non-generic. Although SVS consumes extra energy, they can be easily implemented in short time compared to other two approaches- energy model and RSSI. Looking at the solution space, employing SVS with the capability of activating it at only specific time interval seems to be an efficient solution. Estimating the remaining energy level is only a half solution. Predicting how long the remaining energy will last based on the current functioning of the device is worth researching in the future.

Chapter 8 Conclusions

It is an established fact that IoT is here to stay. A number of nodes that will be connected in the near future is in the tunes of trillions. This growth in IoT is fueled because the bottom-line value (higher revenue at lower cost) that can be created or will migrate among various business domains based on their ability to harness IoT is enormous. Although IoT has many benefits to offer, there are challenges at various levels that pose a serious threat to the seamless adoption of IoT.

To this end, chapter 1 of this thesis provides the broad overview of various concerns that crop up while deploying IoT network. The crucial aspects of IoT are sensing and communication. Achieving high reliability in communication with low latency and energy consumption is not an easy task. Therefore, chapter 1 provides more detailed explanation of technical concerns or challenges related to communication aspect of IoT network and present the research question and objectives that the thesis addresses.

Chapter 2 is a literature review that presents an overview of steps and various approaches to performing Transmission Power Control (TPC). Based on how link quality metrics are calculated, it is broadly classified into hardware and software link quality metrics and explained. The main limitation of link quality metrics is that it is sensitive and reactive in nature. The reason for their sensitiveness and consequences of employing them in TPC algorithms are briefly explained. Various techniques to process link quality metrics are presented in the literature. However, in chapter 2, these techniques are classified into machine learning and non-machine learning approaches and their merits and demerits are briefly discussed.

Environmental conditions such as humidity, temperature, movement of people in the office space are known to abruptly change the value of various link quality metrics such as LQI, RSSI. This frequent fluctuation is recognized by TPC module and it changes the transmission power level with an intention to enhance the communication link quality. However, the impact of this change in the network has not been studied. Therefore, chapter 3 provides experimental evidence on how a single node performing TPC in the network with multiple transmitting nodes negatively influences the performance of the entire network.

In addition, the chapter provides evidence on how TPC module inadvertently influences routing and MAC layers.

The two main goals of TPC is to employ lower transmission power level when the condition in the propagation medium is stable and increase it to the next power level when the communication quality between a pair of nodes fall below a certain predefined threshold. The design decision to use lower transmission power results in two main problems – hidden node and unstable links. These two issues are more prominent in a dense network. Evidently, as the number of nodes increases, hidden node problem increases, and collision arises with it when there are multiple transmitting nodes. Hence, there is an increase in latency and additional energy consuming retransmission. Therefore, the design decision used by older TPC module to employ transmission power based solely on the variation in propagation medium without considering the communication pattern of the neighboring nodes results in low reliability. Older TPC modules increase their transmission power step-wise. However, there is no guarantee that step-wise increment in power level can enhance the link quality. To this end, chapter 4 investigates if there is a trade-off in terms of reliability and energy consumption in the adoption of lower and higher transmission power by the sensor nodes. Based on the results conducted, the chapter proposes to consider employing higher transmission power that eliminates hidden-node problem rather than electing power based on the variation in the propagation medium.

Early prediction of link quality between a pair of nodes is helpful as the reliability of the link can be maintained or enhanced by adjusting the transmission power accordingly beforehand to compensate the degradation in the propagation medium. Although popular machine learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN) etc. could be used for link quality prediction, they require a large amount of training samples for training them before they are capable of predicting. In addition, implementing them in popular embedded IoT devices such as TelosB, MicaZ, and Econotag etc. is challenging due to their hardware limitations. Considering the time required for collecting the data, training the learning model and hardware limitation of IoT devices, learning algorithms are not the best choice.

To this end, chapter 5 investigates other alternative techniques to learning algorithms and provides a comprehensive analysis of prediction accuracy and ease of configuration of some of the many non-machine learning algorithms.

Evaluating the cost involved in terms of operational uptime, ease of configuration and implementation, this chapter proposes to use non-machine learning algorithm such as Weighted Moving Average (WMA) as the best algorithm for implementing proactive TPC.

As variation in the communication medium is frequent, older TPC algorithms use energy inefficient and time consuming initialization phase to get accurate information on the variation in link quality. However, it is known that even a prolonged initialization phase cannot accurately model the variation. In addition, they employ transmission power just enough to connect to its immediate neighbor. These low power links are prone to failure due to internal interference. In addition, it alleviates the hidden-node problem. This causes a collision in the network when there multiple transmitting nodes and hence more packets are retransmitted at the expense of higher energy consumption. As link quality such as RSSI, LQI are sensitive to environmental conditions, employing them as a primary input forces the TPC module to frequently adjust the transmission power causing more collision. In addition, variation in these metrics is specific to a location and hence TPC modules that depend on RSSI, LQI makes them non-generic. This requires time consuming fine-tuning of TPC module when TPC enabled IoT nodes are deployed in a new location. To eliminate the drawbacks present in older TPC model, chapter 6 proposes a new sender-side Data Aware Transmission Power Control (DA-TPC) algorithm that uses priority of the data on per-packet basis rather than sensitive parameters such as RSSI as the decision point to adjust the transmission power. DA-TPC has a shorter initialization phase and selects appropriate transmission power that eliminates the hidden-node problem. Experiments conducted shows that the nodes employing DA-TPC perform better than reactive TPC in terms of Packet Delivery Ratio (PDR), latency and energy consumption. In addition, the results also reveal that frequent fluctuation of transmission power by DA-TPC does not affect the performance of the entire network as it does in the case of reactive TPC.

Generally, transceiver consumes more energy than any other embedded component of a sensor node. Therefore, utilizing the knowledge of remaining energy level as an input to adjust the transmission power level can be helpful. To this end, the last chapter of this thesis explores the applicability of employing RSSI as an input to various state-of-the-art supervised machine-learning algorithms. The experiments conducted shows that the nature of RSSI values can aid the machine learning algorithms to predict the fall in the energy level only when they reach cut-off level (1.6V to 1.5V) –a minimum operational voltage required for the normal operation of the TelosB sensor motes. This suggests that usage of RSSI in predicting the remaining energy level is not possible. To conclude, this thesis started out with following research question mentioned chapter 1

How can we enhance the communication reliability of IoT network with Transmission Power Control (TPC)?

To address this question, first, shortcoming if any, in the traditional receiver based reactive TPC has to be established. This thesis experimentally highlights four major drawbacks such as prolonged initialization phase, hidden-node problem, higher latency, and energy cost with lower reliability, non-generic nature of reactive TPC algorithms. Therefore, older TPC models such as ODTPC, ATPC, and ART cannot enhance the communication reliability. To mitigate the problems encountered with traditional receiver based reactive TPC algorithms, DA-TPC algorithm is proposed. Experiments show that DA-TPC certainly can enhance the reliability of the network with lower latency and transmission energy.

Although the DA-TPC algorithm is beneficial, it is validated only for static sizeable IoT network (e.g.10 nodes network). In addition, the performance of DA-TPC is tested with only one dynamic routing protocol – CTP and MAC protocol – ContikiMAC. Evaluation of the proposed algorithm with different protocols at lower layers is crucial. Furthermore, the performance of reactive TPC model and DA-TPC model is tested in a network with multiple transmitting nodes. The performance of both variants of algorithms in a scheduled network is necessary for widespread adoption of DA-TPC.

References

[1] J. Martinez, The Internet of Things will cause IP traffic to skyrocket 300% by 2018 | TechRadar, (n.d.). http://www.techradar.com/us/news/internet/broadband/the-internet-of-things-will-cause-ip-traffic-to-skyrocket-300-by-2018-1252812 (accessed April 28, 2016).

[2] O. Vermesan, P. Friess, eds., Internet of Things - Global Technological and Societal Trends, River Publishers, 2011.

[3] Cisco, The Internet of Things: At-A-Glance, 2014. http://electronicdesign.com/communications/internet-things-needs-firewalls-too.

[4] D. Wellers, Is this the future of the Internet of Things?, SAP Community Netw. (2015).

[5] John esposito, DZone Guide 2015 The Internet Of Things, 2015. https://dzone.com/guides/internet-of-things-1.

[6] D.H. Joseph Bradley, Joel Barbier, Embracing the Internet of Everything To Capture Your Share of $14.4 Trillion, 2013. http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoE_Economy.pdf (accessed April 29, 2016).

[7] Lopez Research LLC, Building Smarter Manufacturing With The Internet of Things (IoT), San Francisco, CA, 2014. http://cdn.iotwf.com/resources/6/iot_in_manufacturing_january.pdf (accessed May 2, 2016).

[8] O. Chipara, C. Lu, T.C. Bailey, G.-C. Roman, Reliable clinical monitoring using wireless sensor networks, in: Proc. 8th ACM Conf. Embed. Networked Sens. Syst. - SenSys '10, ACM Press, New York, New York, USA, 2010: p. 155. doi:10.1145/1869983.1869999.

[9] N. Meghanathan, S. Boumerdassi, N. Chaki, D. Nagamalai, eds., Recent Trends in Network Security and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-14478-3.

[10] D.R. Raymond, S.F. Midkiff, Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses, IEEE Pervasive Comput. 7 (2008) 74–81. doi:10.1109/MPRV.2008.6.

[11] Willow Technologies, The TelosB Mote Platform, San Jose, California, n.d. http://www.willow.co.uk/html/telosb_mote_platform.php.

[12] MEMSIC Inc, M ICAz Wireless Measurement System Report, San Jose, California, n.d. http://www.memsic.com/wireless-sensor-networks/.

[13] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, et al., Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks, in: 2007 6th Int. Symp. Inf. Process. Sens. Networks, IEEE, 2007: pp. 254–263. doi:10.1109/IPSN.2007.4379685.

[14] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, Wireless sensor networks for habitat monitoring, in: Proc. 1st ACM Int. Work. Wirel. Sens. Networks Appl. - WSNA ’02, ACM Press, New York, New York, USA, 2002: p. 88. doi:10.1145/570738.570751.

[15] C. Albaladejo, P. Sánchez, A. Iborra, F. Soto, J.A. López, R. Torres, Wireless Sensor Networks for oceanographic monitoring: a systematic review., Sensors (Basel). 10 (2010) 6948–68. doi:10.3390/s100706948.

[16] N. Baccour, A. Koubâa, L. Mottola, M.A. Zúñiga, H. Youssef, C.A. Boano, et al., Radio link quality estimation in wireless sensor networks, ACM Trans. Sens. Networks. 8 (2012) 1–33. doi:10.1145/2240116.2240123.

[17] Jin-A Park, Seung-Keun Park, Dong-Ho Kim, Pyung-Dong Cho, Kyoung-Rok Cho, Experiments on radio interference between wireless LAN and other radio devices on a 2.4 GHz ISM band, in: 57th IEEE Semiannu. Veh. Technol. Conf. 2003. VTC 2003-Spring., IEEE, 2003: pp. 1798–1801. doi:10.1109/VETECS.2003.1207133.

[18] L. Mottola, G. Pietro Picco, M. Ceriotti, Ş. Gună, A.L. Murphy, Not all wireless sensor networks are created equal, ACM Trans. Sens. Networks. 7 (2010) 1–33. doi:10.1145/1824766.1824771.

[19] O. Incel, S. Dulman, P. Jansen, S. Mullender, Multi-Channel Interference Measurements for Wireless Sensor Networks, in: Proceedings. 2006 31st IEEE Conf. Local Comput. Networks, IEEE, 2006: pp. 694–701. doi:10.1109/LCN.2006.322179.

[20] S. Kostin, R.M. Salles, C.L. de Amorim, An approach for wireless sensor networks topology control in indoor scenarios, Proc. 5th Int. Lat. Am. Netw. Conf. - LANC ’09. (2009) 1. doi:10.1145/1636682.1636684.

[21] K. Bannister, G. Giorgetti, S.K. Gupta, Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization, in: Proc. 5th Work. Embed. Networked Sensors (HotEmNets 08), 2008.

[22] J. Thelen, Radio Wave Propagation in Potato Fields, in: Proc. First Work. Wirel. Netw. Meas. - WiNMee 2005, Trentino, Italy, 2004: p. 5.

[23] M. Zennaro, H. Ntareme, A. Bagula, Experimental evaluation of temporal and energy characteristics of an outdoor sensor network, Proc. Int. Conf. Mob. Technol. Appl. Syst. - Mobil. ’08. (2008) 5.

doi:10.1145/1506270.1506391.

[24] C. DataLoggers, The Basics of Signal Attenuation, Ohio, US, 2012. http://www.dataloggerinc.com/content/files/whitepaper/wireless_data_loggers_accsense_temperature_datalogger.pdf (accessed May 10, 2016).

[25] N. Amini, M. Sarrafzadeh, Experimental analysis of IEEE 802.15.4 for on/off body communications, in: 2011 IEEE 22nd Int. Symp. Pers. Indoor Mob. Radio Commun., IEEE, 2011: pp. 2138–2142. doi:10.1109/PIMRC.2011.6139893.

[26] A. Saipulla, B. Liu, J. Wang, Barrier coverage with airdropped wireless sensors, in: MILCOM 2008 - 2008 IEEE Mil. Commun. Conf., IEEE, 2008: pp. 1–7. doi:10.1109/MILCOM.2008.4753650.

[27] M. Wadhwa, V. Rali, S. Shetty, The impact of antenna orientation on wireless sensor network performance, in: 2009 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol., IEEE, 2009: pp. 143–147. doi:10.1109/ICCSIT.2009.5234978.

[28] Q. Wang, Traffic Analysis & Modeling in Wireless Sensor Networks and Their Applications on Network Optimization and Anomaly Detection, Netw. Protoc. Algorithms. 2 (2010) 74–92. doi:10.5296/npa.v2i1.328.

[29] J. Jeong, D. Culler, J.-H. Oh, Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks, in: 2007 Fourth Int. Conf. Networked Sens. Syst., IEEE, 2007: pp. 27–34. doi:10.1109/INSS.2007.4297383.

[30] D. Lymberopoulos, Q. Lindsey, A. Savvides, Wireless Sensor Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi:10.1007/11669463.

[31] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall PTR, 2001. http://dl.acm.org/citation.cfm?id=559977 (accessed May 10, 2016).

[32] A. Forster, D. Puccinelli, S. Giordano, Sensor node lifetime: An experimental study, in: 2011 IEEE Int. Conf. Pervasive Comput. Commun. Work. (PERCOM Work., IEEE, 2011: pp. 202–207. doi:10.1109/PERCOMW.2011.5766869.

[33] G. Hackmann, O. Chipara, C. Lu, Robust topology control for indoor wireless sensor networks, Proc. SenSys. (2008) 1–14. doi:10.1145/1460412.1460419.

[34] I. Broustis, J. Eriksson, S. V Krishnamurthy, M. Faloutsos, Implications of Power Control in Wireless Networks: A Quantitative Study, Pam. 4427 (2007) 83–93.

[35] P. Santi, Topology control in wireless ad hoc and sensor networks, ACM Comput. Surv. 37 (2005) 164–194. doi:10.1145/1089733.1089736.

[36] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: Proc. 7th ACM Conf. Embed. Networked Sens. Syst. - SenSys '09, ACM Press, New York, New York, USA, 2009: p. 1. doi:10.1145/1644038.1644040.

[37] A. Dunkels, B. Gronvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: 29th Annu. IEEE Int. Conf. Local Comput. Networks, IEEE (Comput. Soc.), 2004: pp. 455–462. doi:10.1109/LCN.2004.38.

[38] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-Level Sensor Network Simulation with COOJA, in: Proceedings. 2006 31st IEEE Conf. Local Comput. Networks, IEEE, 2006: pp. 641–648. doi:10.1109/LCN.2006.322172.

[39] J.T. Adams, An Introduction to IEEE STD 802.15.4, in: 2006 IEEE Aerosp. Conf., IEEE, 2006: pp. 1–8. doi:10.1109/AERO.2006.1655947.

[40] R. Horak, Webster's New World Telecom Dictionary, 1st ed., Webster's New World, 2007. http://www.yourdictionary.com/convergence-sublayer (accessed May 19, 2016).

[41] W.-B. Pottner, L. Wolf, Probe-Based Transmission Power Control for Dependable Wireless Sensor Networks, in: 2013 IEEE Int. Conf. Distrib. Comput. Sens. Syst., IEEE, 2013: pp. 44–51. doi:10.1109/DCOSS.2013.71.

[42] A. Cerpa, J.L. Wong, M. Potkonjak, D. Estrin, Temporal properties of low power wireless links, in: Proc. 6th ACM Int. Symp. Mob. Ad Hoc Netw. Comput. - MobiHoc '05, ACM Press, New York, New York, USA, 2005: p. 414. doi:10.1145/1062689.1062741.

[43] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian, Measurement and characterization of link quality metrics in energy constrained wireless sensor networks, in: GLOBECOM '03. IEEE Glob. Telecommun. Conf. (IEEE Cat. No.03CH37489), IEEE, 2003: pp. 446–452. doi:10.1109/GLOCOM.2003.1258278.

[44] K.-H. Kim, K.G. Shin, On accurate measurement of link quality in multi-hop wireless mesh networks, in: Proc. 12th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '06, ACM Press, New York, New York, USA, 2006: p. 38. doi:10.1145/1161089.1161095.

[45] M. Caleffi, L. Paura, Bio-inspired link quality estimation for wireless mesh networks, in: 2009 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks Work., IEEE, 2009: pp. 1–6. doi:10.1109/WOWMOM.2009.5282423.

[46] roshan kotian, G. Exarchakos, A. Liotta, Assessment of Proactive Transmission Power Control for Wireless Sensor Networks, in: Proc. 9th Int. Conf. Body Area Networks, ICST, 2014: pp. 253–259.

doi:10.4108/icst.bodynets.2014.258209.

[47] IEEE Standard for Information Technology, (2003) 0_1–670. doi:10.1109/IEEESTD.2003.94389.

[48] Y.-D. Young-Dong Lee, D.-U. Do-Un Jeong, H.-J. Hoon-Jae Lee, Performance analysis of wireless link quality in wireless sensor networks, in: 5th Int. Conf. Comput. Sci. Converg. Inf. Technol., IEEE, 2010: pp. 1006–1010. doi:10.1109/ICCIT.2010.5711208.

[49] H. Ali, A Performance Evaluation of RPL in Contiki, Blekinge Institute of Technology,Sweden, 2012. http://www.medieteknik.bth.se/fou/cuppsats.nsf/all/418758357c3441b8c1257aca002d2932/$file/BTH2012Hazrat.pdf.

[50] F. Pianegiani, miniTP: A protocol for the minimization of the transmit power in wireless networks, Instrum. Meas. Technol. Conf. (I2MTC), 2011 IEEE. (2011) 1–6. doi:10.1109/IMTC.2011.5944355.

[51] D.S.J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proc. 9th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom ’03, ACM Press, New York, New York, USA, 2003: p. 134. doi:10.1145/938985.939000.

[52] C.E. Koksal, H. Balakrishnan, Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks, IEEE J. Sel. Areas Commun. 24 (2006) 1984–1994. doi:10.1109/JSAC.2006.881637.

[53] C.E. Koksal, H. Balakrishnan, Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks, IEEE J. Sel. Areas Commun. 24 (2006) 1984–1994. doi:10.1109/JSAC.2006.881637.

[54] D. Son, Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks, First Annu. IEEE Commun. Soc. Conf. Sens. Ad Hoc Commun. Networks. 00 (2004) 289–298. doi:10.1109/SAHCN.2004.1381929.

[55] G. Hackmann, C. Lu, Practical control of transmission power for Wireless Sensor Networks, in: 2012 20th IEEE Int. Conf. Netw. Protoc., IEEE, 2012: pp. 1–10. doi:10.1109/ICNP.2012.6459981.

[56] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, E.J. Coyle, A Kalman Filter Based Link Quality Estimation Scheme for Wireless Sensor Networks, in: IEEE GLOBECOM 2007-2007 IEEE Glob. Telecommun. Conf., IEEE, 2007: pp. 875–880. doi:10.1109/GLOCOM.2007.169.

[57] J. Kim, S. Chang, Y. Kwon, ODTPC: On-demand Transmission Power Control for Wireless Sensor Networks, in: 2008 Int. Conf. Inf. Netw. - ICOIN 2008, IEEE, Busan, 2008: pp. 1–5. doi:10.1109/ICOIN.2008.4472775.

[58] M.M.Y. Masood, G. Ahmed, N.M. Khan, A Kalman filter based

adaptive on demand transmission power control (AODTPC) algorithm for wireless sensor networks, 2012 Int. Conf. Emerg. Technol. (2012) 1–6. doi:10.1109/ICET.2012.6375499.

[59] S.-H. Oh, TPC-BS: Transmission power control based on binary search in the wireless sensor networks, 2012 IEEE Sensors Appl. Symp. Proc. (2012) 1–6. doi:10.1109/SAS.2012.6166321.

[60] T. Christiano Silva, L. Zhao, Machine Learning in Complex Networks, Springer International Publishing AG Switzerland, 2016. doi:10.1007/978-3-319-17290-3.

[61] Y. Wang, M. Martonosi, L.-S. Peh, Predicting link quality using supervised learning in wireless sensor networks, ACM SIGMOBILE Mob. Comput. Commun. Rev. 11 (2007) 71. doi:10.1145/1317425.1317434.

[62] G.A. Di Caro, M. Kudelski, E.F. Flushing, J. Nagi, I. Ahmed, L.M. Gambardella, Online supervised incremental learning of link quality estimates in wireless networks, in: 2013 12th Annu. Mediterr. Ad Hoc Netw. Work., IEEE, 2013: pp. 133–140. doi:10.1109/MedHocNet.2013.6767422.

[63] T. Liu, A.E. Cerpa, Foresee (4C): Wireless link prediction using link features, in: Inf. Process. 10th Int. Conf. onSensor Networks (IPSN), 2011 , IEEE, Chicago, IL , 2011: pp. 294–305.

[64] L. Liu, C. Zang, J. Shu, Y. Ge, Y. Zhou, A Link Quality Assessment Model for WSNs Based on BDCT-SVM, in: Springer Berlin Heidelberg, 2012: pp. 121–127. doi:10.1007/978-3-642-28655-1_19.

[65] R. Kazemi, R. Vesilo, E. Dutkiewicz, R.P. Ren Ping Liu, Reinforcement learning in power control games for internetwork interference mitigation in Wireless Body Area Networks, in: 2012 Int. Symp. Commun. Inf. Technol., IEEE, 2012: pp. 256–262. doi:10.1109/ISCIT.2012.6380902.

[66] R. Maheshwari, S. Jain, S.R. Das, A measurement study of interference modeling and scheduling in low-power wireless networks, in: Proc. 6th ACM Conf. Embed. Netw. Sens. Syst. - SenSys ’08, ACM Press, New York, New York, USA, 2008: p. 141. doi:10.1145/1460412.1460427.

[67] P.L. Kannan Srinivasan, RSSI is Under Appreciated, in: Proc. Third Work. Embed. Networked Sensors - EmNets, 2006: p. 5. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.5362 (accessed May 21, 2014).

[68] S. Lin, J. Zhang, G. Zhou, L. Gu, J.A. Stankovic, T. He, ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks, in: Proc. 4th Int. Conf. Embed. Networked Sens. Syst. - SenSys ’06, ACM Press, New York, New York, USA, 2006: pp. 223–236. doi:10.1145/1182807.1182830.

[69] K. Benkic, M. Malajner, P. Planinsic, Z. Cucej, Using RSSI value for distance estimation in wireless sensor networks based on ZigBee, in: 2008 15th Int. Conf. Syst. Signals Image Process., IEEE, 2008: pp. 303–306. doi:10.1109/IWSSIP.2008.4604427.

[70] S. Hara, D. Anzai, Comparison of Three Estimation Methods for RSSI-Based Localization with Multiple Transmit Antennas, in: 2007 IEEE Internatonal Conf. Mob. Adhoc Sens. Syst., IEEE, 2007: pp. 1–3. doi:10.1109/MOBHOC.2007.4428599.

[71] Chipcon, 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, 2004. http://www.alldatasheet.com/datasheet-pdf/pdf/125399/ETC1/CC2420.html.

[72] Y. Chen, A. Terzis, On the Mechanisms and Effects of Calibrating RSSI Measurements for 802.15.4 Radios, in: Springer Berlin Heidelberg, 2010: pp. 256–271. doi:10.1007/978-3-642-11917-0_17.

[73] Texas Instruments, 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, Dallas, Texas, 2014.

[74] Atmel, Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications, 2009.

[75] V. Shrivastava, D. Agrawal, A. Mishra, S. Banerjee, T. Nadeem, Understanding the limitations of transmit power control for indoor wlans, in: Proc. 7th ACM SIGCOMM Conf. Internet Meas. - IMC ’07, ACM Press, New York, New York, USA, 2007: p. 351. doi:10.1145/1298306.1298356.

[76] A. Dunkels, The ContikiMAC Radio Duty Cycling Protocol, Swedish Institute of Computer Science, 2011. http://soda.swedish-ict.se/5128/1/contikimac-report.pdf (accessed October 27, 2015).

[77] D.S.J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proc. 9th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom ’03, ACM Press, New York, New York, USA, 2003: p. 134. doi:10.1145/938985.939000.

[78] P. Peng Jiang, Q. Qingbo Huang, J. Jianzhong Wang, X. Xiaohua Dai, R. Ruizhong Lin, Research on Wireless Sensor Networks Routing Protocol for Wetland Water Environment Monitoring, in: First Int. Conf. Innov. Comput. Inf. Control - Vol. I, IEEE, 2006: pp. 251–254. doi:10.1109/ICICIC.2006.508.

[79] A. Cerpa, J.L. Wong, L. Kuang, M. Potkonjak, D. Estrin, Statistical model of lossy links in wireless sensor networks, in: IPSN 2005. Fourth Int. Symp. Inf. Process. Sens. Networks, 2005., IEEE, 2005: pp. 81–88. doi:10.1109/IPSN.2005.1440900.

[80] T. Liu, A.E. Cerpa, Temporal Adaptive Link Quality Prediction with Online Learning, ACM Trans. Sens. Networks. 10 (2014) 1–41.

doi:10.1145/2594766.

[81] D.D. Mai, A.T. Tran, Myung Kyun Kim, Measuring link quality based on ETX metric in multi- hop wireless networks, Adv. Sci. Technol. Lett. 46 (2014) 115–118.

[82] N. Javaid, A. Javaid, I.A. Khan, K. Djouani, Performance study of ETX based wireless routing metrics, in: 2009 2nd Int. Conf. Comput. Control Commun., IEEE, 2009: pp. 1–7. doi:10.1109/IC4.2009.4909163.

[83] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, E.J. Coyle, A Kalman Filter Based Link Quality Estimation Scheme for Wireless Sensor Networks, in: IEEE GLOBECOM 2007-2007 IEEE Glob. Telecommun. Conf., IEEE, 2007: pp. 875–880. doi:10.1109/GLOCOM.2007.169.

[84] R. Kotian, G. Exarchakos, D.C. Mocanu, A. Liotta, Predicting Battery Depletion of Neighboring Wireless Sensor Nodes, in: Proc. 13th Int. Conf. Algorithms Archit. Parallel Process. - Vol. 8286, Springer-Verlag New York, Inc., 2013: pp. 276–284. doi:10.1007/978-3-319-03889-6_32.

[85] B. Krishnapuram, Y. Shipeng, B. Rao, eds., Cost-Sensitive Machine Learning, Chapman & Hall/CRC, 2012.

[86] J. Read, A. Bifet, B. Pfahringer, G. Holmes, Batch-Incremental versus Instance-Incremental Learning in Dynamic and Evolving Data, in: Adv. Intell. Data Anal. XI, Springer Berlin Heidelberg, 2012: pp. 313–323. doi:10.1007/978-3-642-34156-4_29.

[87] A. Medela, B. Cendon, L. Gonzalez, R. Crespo, I. Nevares, Future Network and Mobile Summit (FutureNetworkSummit), 2013, Futur. Netw. Mob. Summit (FutureNetworkSummit), 2013. (2013) 1–10.

[88] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, Ad Hoc Networks. 7 (2009) 537–568. doi:10.1016/j.adhoc.2008.06.003.

[89] H. Miao, Y. Jin, Q. Ge, Y. He, Expected Transmission Energy Routing Metric for Wireless Mesh Senor Network, in: 2010 Int. Conf. Comput. Intell. Softw. Eng., IEEE, 2010: pp. 1–4. doi:10.1109/WICOM.2010.5601407.

[90] X. Zhang, Q. Liu, D. Shi, Y. Liu, X. Yu, An Average Link Interference-Aware Routing Protocol for Mobile Ad Hoc Networks, in: 2007 Third Int. Conf. Wirel. Mob. Commun., IEEE, 2007: pp. 10–10. doi:10.1109/ICWMC.2007.18.

[91] M. Caleffi, G. Ferraiuolo, L. Paura, Augmented Tree-based Routing Protocol for Scalable Ad Hoc Networks, in: 2007 IEEE Internatonal Conf. Mob. Adhoc Sens. Syst., IEEE, 2007: pp. 1–6. doi:10.1109/MOBHOC.2007.4428727.

[92] I. Radoi, A. Shenoy, D. Arvind, Evaluation of Routing Protocols for Internet-Enabled Wireless Sensor Networks, in: 2012 8th Int. Conf. Wirel. Mob. Commun., 2012: pp. 56–61. http://www.thinkmind.org/index.php?view=article&articleid=icwmc_2012_3_30_20243.

[93] J. Haerri, Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns, in: Med-Hoc-Net 2006, 5th IFIP Mediterr. Ad-Hoc Netw. Work., IFIP, Italy, 2006: pp. 1 – 8. http://www.eurecom.fr/publication/1952 (accessed April 4, 2016).

[94] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, et al., Real-time Power-Aware Routing in Sensor Networks, in: 200614th IEEE Int. Work. Qual. Serv., IEEE, 2006: pp. 83–92. doi:10.1109/IWQOS.2006.250454.

[95] K. Witheephanich, M.J. Hayes, On the applicability of model predictive power control to an IEEE 802.15.4 wireless sensor network, in: IET Irish Signals Syst. Conf. (ISSC 2009), IET, 2009: pp. 32–32. doi:10.1049/cp.2009.1709.

[96] V. Shrivastava, D. Agrawal, A. Mishra, S. Banerjee, T. Nadeem, Understanding the limitations of transmit power control for indoor wlans, in: Proc. 7th ACM SIGCOMM Conf. Internet Meas. - IMC '07, ACM Press, New York, New York, USA, 2007: p. 351. doi:10.1145/1298306.1298356.

[97] M. Michel, B. Quoitin, Technical Report : ContikiMAC vs X-MAC performance analysis, 2015. http://arxiv.org/abs/1404.3589.

[98] L. (Eds. . Langendoen, K., Hu, W., Ferrari, F., Zimmerling, M., Mottola, ed., Real-World Wireless Sensor Networks - Proceedings of the 5th | Koen Langendoen | Springer, Springer International Publishing, Como, Italy, 2014. doi:10.1007/978-3-319-03071-5.

[99] A. Dunkels, Rime — A Lightweight Layered Communication Stack for Sensor Networks, in: Proc. Eur. Conf. Wirel. Sens. Networks, Springer, LNCS, Delft, The Netherlands, 2007: pp. 1–2. http://dunkels.com/adam/dunkels07rime.pdf (accessed April 17, 2015).

[100] R. Kotian, G. Exarchakos, A. Liotta, Assessment of Proactive Transmission Power Control for Wireless Sensor Networks, in: Proc. 9th Int. Conf. Body Area Networks, ICST, 2014: pp. 253–259. doi:10.4108/icst.bodynets.2014.258209.

[101] M. Chincoli, A. Syed, G. Exarchakos, A. Liotta, Power Control in Wireless Sensor Networks with Variable Interference, Mob. Inf. Syst. (2016).

[102] N. a Pantazis, S. a Nikolidakis, D.D. Vergados, S. Member, Energy-Ef fi cient Routing Protocols in Wireless Sensor Networks : A Survey, IEEE

Commun. Surv. Tutorials. 15 (2013) 551–591. doi:10.1109/SURV.2012.062612.00084.

[103] L.H. a Correia, J.M.S. Nogueira, Transmission power control techniques for MAC protocols in wireless sensor networks, Netw. Oper. Manag. Symp. 2008. NOMS 2008. IEEE. (2008) 1049–1054. doi:10.1109/noms.2008.4575277.

[104] S. Galzarano, C. Savaglio, A. Liotta, G. Fortino, Gossiping-Based AODV for Wireless Sensor Networks, in: 2013 IEEE Int. Conf. Syst. Man, Cybern., IEEE, 2013: pp. 26–31. doi:10.1109/SMC.2013.12.

[105] G. Di Fatta, S. Gaglio, G. Lo Re, M. Ortolani, Adaptive routing in active networks, in: OPENARCH 2000 Third IEEE Conf. Open Archit. Netw. Program., Tel-Aviv, Israel, 2000: pp. 8–11. http://centaur.reading.ac.uk/6143/1/2000_DiFatta-openarch2000.pdf (accessed December 12, 2015).

[106] G. Exarchakos, I. Oztelcan, D. Sarakiotis, A. Liotta, plexi: Adaptive re-scheduling web service of time synchronized low-power wireless networks, J. Netw. Comput. Appl. (2016).

[107] L. Xu, D.T. Delaney, G.M.P. O'Hare, R. Collier, The impact of transmission power control in wireless sensor networks, Netw. Comput. Appl. (NCA), 2013 12th IEEE Int. Symp. 63 (2013) 255–258.

[108] M.N. Jambli, H. Lenando, K. Zen, S.M. Suhaili, A. Tully, The Effects of Transmission Power Control in Mobile Ad-Hoc Sensor Networks, Procedia Eng. 41 (2012) 1244–1252. doi:http://dx.doi.org/10.1016/j.proeng.2012.07.307.

[109] J.-P. Sheu, K.-Y. Hsieh, Y.-K. Cheng, Distributed transmission power control algorithm for wireless sensor networks, J. Inf. Sci. Eng. 1463 (2009) 1447–1463.

[110] A.R. Kotian, G. Exarchakos, S. Stavros, A. Liotta, Impact of Transmission Power Control in Multi - hop Networks, Futur. Gener. Comput. Syst. J. (2016) 26.

[111] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, et al., Real-time Power-Aware Routing in Sensor Networks, in: 200614th IEEE Int. Work. Qual. Serv., IEEE, 2006: pp. 83–92. doi:10.1109/IWQOS.2006.250454.

[112] T.A. Tian He, Pascal Vicaire, Ting Yan, Liqian Luo, Lin Gu, Gang Zhou, Radu Stoleru, Qing Cao, John A. Stankovic, Achieving real-time target tracking using wireless sensor networks, in: Twelfth IEEE Real-Time Embed. Technol. Appl. Symp., IEEE Comput. Soc, 2006: pp. 37–48. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.6728 (accessed April 18, 2016).

[113] O. Chipara, S. Bhattacharya, A Spatiotemporal Query Service for

Mobile Users in Sensor Networks, in: 25th IEEE Int. Conf. Distrib. Comput. Syst., IEEE, 2005: pp. 381–390. doi:10.1109/ICDCS.2005.7.

[114] S.K. Das, Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution, IEEE Trans. Parallel Distrib. Syst. 19 (2008) 710–720. doi:10.1109/TPDS.2007.70770.

[115] U. Pešović, J. Mohorko, K. Benkič, Ž. Čučej, Effect of hidden nodes in IEEE 802.15.4/ZigBee Wireless Sensor Networks, in: Proc. 17th Telecommun. Forum, Serbia, Belgrade, 2009: p. 4.

[116] S. Vijayakumar, S. Schaal, Locally Weighted Projection Regression: An O(n) Algorithm for Incremental Real Time Learning in High dimensional Space, in: Seventeenth Int. Conf. Mach. Learn. (ICML 2000), 2000: pp. 1079–1086.

[117] H.H.W.J. Bosman, A. Liotta, G. Iacca, H.J. Wortche, Anomaly Detection in Sensor Systems Using Lightweight Machine Learning, in: 2013 IEEE Int. Conf. Syst. Man, Cybern., IEEE, 2013: pp. 7–13. doi:10.1109/SMC.2013.9.

[118] I. Rish, An empirical study of the naive bayes classifier, (2001).

[119] D. Nguyen, T. Ho, Speeding-up Model Selection for Support Vector Machines, in: FLAIRS Conf., 2005: p. 6.

[120] M.A. Alsheikh, S. Lin, D. Niyato, H.-P. Tan, Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications, IEEE Commun. Surv. Tutorials. 16 (2014) 1996–2018. doi:10.1109/COMST.2014.2320099.

[121] M. Zuniga, B. Krishnamachari, Analyzing the transitional region in low power wireless links, in: SECON 2004., IEEE, n.d.: pp. 517–526. doi:10.1109/SAHCN.2004.1381954.

[122] Torben Grøn Helligsø, RSSI and LQI vs . Distance Measurement, Aarhus School Of Engineering, Denmark, 2011.

[123] C.C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, Int. J. Forecast. 20 (2004) 5–10. doi:10.1016/j.ijforecast.2003.09.015.

[124] K. Pearson, Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 187 (1896) 253–318. doi:10.1098/rsta.1896.0007.

[125] R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems 1, Trans. ASME – J. Basic Eng. 82 (1960) 35–45.

[126] J.M. Lucas, M.S. Saccucci, Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements, Technometrics. 32 (1990) 1–12. doi:10.1080/00401706.1990.10484583.

[127] A. Bifet, Adaptive learning and mining for data streams and frequent patterns, ACM SIGKDD Explor. Newsl. 11 (2009) 55.

doi:10.1145/1656274.1656287.

[128] M.M.Y. Masood, G. Ahmed, N.M. Khan, Modified on demand transmission power control for wireless sensor networks, in: 2011 Int. Conf. Inf. Commun. Technol., IEEE, 2011: pp. 1–6. doi:10.1109/ICICT.2011.5983544.

[129] S. Ben Othman, A. Trad, H. Youssef, Performance evaluation of encryption algorithm for wireless sensor networks, in: 2012 Int. Conf. Inf. Technol. E-Services, IEEE, 2012: pp. 1–8. doi:10.1109/ICITeS.2012.6216690.

[130] C. Ma, Y. Yang, Battery-aware routing for streaming data transmissions in wireless sensor networks, in: 2nd Int. Conf. Broadband Networks, 2005., IEEE, 2005: pp. 499–508. doi:10.1109/ICBN.2005.1589650.

[131] D. Panigrahi T, D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, et al., Battery life estimation of mobile embedded systems, in: VLSI Des. 2001. Fourteenth Int. Conf. VLSI Des., IEEE Comput. Soc, 2001: pp. 57–63. doi:10.1109/ICVD.2001.902640.

[132] D. Rakhmatov, S. Vrudhula, Energy management for battery-powered embedded systems, ACM Trans. Embed. Comput. Syst. 2 (2003) 277–324. doi:10.1145/860176.860179.

[133] S. Vrudhula, D.N. Rakhmatov, Battery modeling for energy-aware system design, Computer (Long. Beach. Calif). 36 (2003) 77–87. doi:10.1109/MC.2003.1250886.

[134] C. Behrens, O. Bischoff, M. Lueders, R. Laur, Energy-efficient topology control for wireless sensor networks using online battery monitoring, Adv. Radio Sci. 5 (2007) 205–208. doi:10.5194/ars-5-205-2007.

[135] E. Nataf, O. Festor, Online Estimation of Battery Lifetime for Wireless Sensor Network, 2012.

[136] B. Tong, G. Wang, W. Zhang, C. Wang, Node Reclamation and Replacement for Long-Lived Sensor Networks, IEEE Trans. Parallel Distrib. Syst. 22 (2011) 1550–1563. doi:10.1109/TPDS.2011.25.

[137] M. Mafuta, M. Zennaro, A. Bagula, G. Ault, H. Gombachika, T. Chadza, Successful deployment of a Wireless Sensor Network for precision agriculture in Malawi, in: 2012 IEEE 3rd Int. Conf. Networked Embed. Syst. Every Appl., IEEE, 2012: pp. 1–7. doi:10.1109/NESEA.2012.6474009.

[138] I.H. Yano, V.C. Oliveira, E. Alberto, D.M. Fagotto, A.D.A. Mota, L. Toledo, et al., PREDICTING BATTERY CHARGE DEPLETION IN WIRELESS SENSOR NETWORKS USING RECEIVED SIGNAL STRENGTH INDICATOR, J. Comput. Sci. 9 (2013) 821–826. doi:10.3844/jcssp.2013.821.826.

[139] A. Bifet, R. Gavalda, Learning from Time-Changing Data with Adaptive

Windowing., SDM. 7 (2007) 2007.

[140] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci., IEEE Comput. Soc, 2000: p. 10. doi:10.1109/HICSS.2000.926982.

[141] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, An empirical study of low-power wireless, ACM Trans. Sens. Networks. 6 (2010) 1–49. doi:10.1145/1689239.1689246.

[142] G. Zhou, T. He, S. Krishnamurthy, J.A. Stankovic, Models and solutions for radio irregularity in wireless sensor networks, ACM Trans. Sens. Networks. 2 (2006) 221–262. doi:10.1145/1149283.1149287.

[143] A. Boschetti, L. Massaron, Scikit-learn: Machine Learning in Python, Packt Publishing Ltd., 2015. http://jmlr.org/papers/v12/pedregosa11a.html.

[144] V.V. B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, Comparing support vector machines with gaussian kernels to radial basis function classifiers, IEEE Trans. Signal Process. 45 (1997) 2758–2765.

[145] G. Hackeling, Mastering Machine Learning with scikit-learn Apply effective learning algorithms to real-world problems using scikit-learn Mastering Machine Learning with scikit-learn Cover image, Packt Publishing Ltd., 2014. www.packtpub.com.

[146] B. Wang, C.M. Wong, F. Wan, P.U. Mak, P.I. Mak, M.I. Vai, Gaussian mixture model based on genetic algorithm for brain-computer interface, in: 2010 3rd Int. Congr. Image Signal Process., IEEE, 2010: pp. 4079–4083. doi:10.1109/CISP.2010.5646204.

[147] L. Breiman, Random Forests, Mach. Learn. 45 (2001) 5–32. doi:10.1023/A:1010933404324.

[148] M. Kirk, Thoughtful Machine Learning, O’Reilly Media, 2015.

[149] G.A.F. Seber, A.J. Lee, Linear Regression Analysis, 2 edition, Wiley, 2003.

[150] NXP Laboratories UK Ltd, Co-existence of IEEE 802.15.4 at 2.4 GHz Application Note, UK, 2013.

List of Publications

Journals

1. R. Kotian, G. Exarchakos, S. Stavros, A. Liotta, "Impact of Transmission Power Control in Multi - hop Networks", Submitted to Futur. Gener. Comput. Syst. J. (2016) 26.

International Conferences

2. R. Kotian, G. Exarchakos, D.C. Mocanu, A. Liotta, "Predicting Battery Depletion of Neighboring Wireless Sensor Nodes," submitted to 13th International Conference on Algorithms Architecture and Parallel Process, 2012.

3. R Kotian, G.Exarchakos, A.Liotta, "Assessment of Proactive Transmission Power Control for Wireless Sensor Networks" submitted to 9th International Conference on Body Area Networks ICST, 2014.

4. R.Kotian, G. Exarchakos, A. Liotta, "Data Driven Transmission Power Control for Wireless Sensor Networks (DA-TPC) submitted to 8th International Conference on Internet and Distributed Computing Systems, IDCS, 2015.

5. R.Kotian, G.Exarchakos, A.Liotta, "Reliable low-power wireless networks over unstable transmission power" submitted to, 14th IEEE International Conference on Networking, Sensing, and Control, ICNSC, 2017.

Book Chapters

6. R. Kotian, G. Exarchakos, A. Liotta, A. Liotta "Run-time Validation Framework" submitted to, Runtime Reconfiguration in Network Embedded Systems - Design and Testing Practices, Springer Publication, 2016.

7. Coen van Leeuwen, Vicente Hernández Díaz, R. Kotian, et al “An Illustrative Application Example: Cargo State Monitoring” submitted to, Runtime Reconfiguration in Network Embedded Systems - Design and Testing Practices, Springer Publication, 2016.

Curriculum Vitae

Roshan Kotian was born in Manipal, India, on July 8, 1982. He earned his Bachelors of Science (BSc) degree in Computer Science from Mangalore University, India. In November 2007, he received his Masters of Computer Applications degree from Manipal Institute of Technology, India.

After his studies, he worked as Associate Software Engineer with Collabera Solutions Pvt. Ltd. for two years. From 2008 to 2010, he served as associate Professor in Shri Devi Institute of Technology, India. He joined Technical University of Eindhoven (TU/e) and earned his Professional Doctorate in Software Engineering (PDEng) in 2012. His PDEng thesis was carried out in Embedded Systems Institute (ESI) and was entitled ‘Simulation Environment for Maritime Safety and Security Systems’.

In October 2012, Roshan worked as a Ph.D. researcher in the Electro-Optical Communications group, department of Electrical Engineering, Eindhoven University of Technology, under the supervision of Prof. dr. Antonio Liotta

Roshan’s research was conducted under the framework of the European Union ARTEMIS project, DEMANES (Design, Monitoring and Operation of Adaptive Networked Embedded Systems). His research activities were focused on Wireless Sensor Network, Embedded devices and application of Machine Learning in Internet-of-Things.

Since Oct 2016, Roshan is a member of Bright Society and has been working as Senior Software Engineer in TomTom Data B.V Eindhoven, The Netherlands.