Spectrum-aware Adaptive Communication for the Internet of Things

Peng Du

January 2014

A Dissertation Submitted to
Birkbeck College, University of London
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

School of Computer Science & Information Systems
Birkbeck College
University of London

Declaration

This thesis is the result of my own work, except where explicitly acknowledged in the text.

Peng Du _____

January 16, 2014

Abstract

The *Internet of Things* (IoT) is widely expected to be the next stage of development for the Internet and brings together material artefacts and their information representations. One of its enabling infrastructures is low-power wireless networks which due to their restricted capacities are especially sensitive to ambient noise from co-located wireless devices utilising the same band. This thesis investigates mechanisms for enhancing lower-power wireless network communications in the presence of ambient noise. To this end, we extend the channel hopping technique of IEEE 802.15.4.e, proposing the *Adaptive Time-slotted Channel Hopping* (A-TSCH) framework. A core ingredient of A-TSCH is the use of *blacklisting* to eliminate noisy channels from the channel hopping sequence. This approach employs knowledge derived from periodically probing the channel noise floor. Blacklists are generated through a series of cooperative processes among multiple nodes to account for spatial diversity of spectral condition. Cross-layer optimisations between A-TSCH and IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) are also investigated. A new routing metric *Spectral Link Cost* (SLC) is introduced to reflect A-TSCH channel information thus incorporating spectral awareness to routing decisions. We evaluate the performance of A-TSCH and SLC through simulations and realistic deployment using motes. We find that the proposed framework and metric support improved communication performance when compared against existing techniques.

Publication

Publications Related to this thesis:

Peng Du; Roussos, G., "Adaptive channel hopping for wireless sensor networks," Mobile and Wireless Networking (iCOST), 2011 International Conference on Selected Topics in, pp.19,23, 10-12 Oct. 2011, Shanghai, P.R. China

Peng Du; Roussos, G., "Adaptive time slotted channel hopping for wireless sensor networks," Computer Science and Electronic Engineering Conference (CEEC), 2012 4th, pp.29,34, 12-13 Sept. 2012, Essex, UK

Du P, Roussos G. Adaptive Communication Techniques for the Internet of Things. Journal of Sensor and Actuator Networks. 2013; 2(1):122-155.

Peng Du; Roussos, G., "Spectrum-Aware Wireless Sensor Networks," IEEE PIMRC 2013, 8-11 September, 2013, London, UK.

To my family

Acknowledgements

My deepest gratitude to my supervisor Prof. George Roussos who has expertly and patiently steering me throughout my research with profound knowledge in related areas. I also sincerely appreciate the insightful critiques and constructive recommendations from Prof. Kun Yang and Dr. Vladimir Dyo that help the analysis and further improvement of this work.

Massive thanks to my departmental colleagues whom I had the privilege to work with as teaching assistant, which has been an inspiring and memorable experience. I would also like to thank fellow researchers for our enjoyable discussions and friendship. In particular, Michalis Zoumboulakis has been so kind to offer help at different stages of my work which is truly appreciated.

To my parents JianHua Du and Yanyan Ren, I would like to acknowledge my eternal indebtedness for it is their unconditional faith and support that enabled me to carry through this work. And my warmest affection to my fiancee, Hui Li, whose unwavering love has kept me uplifted even when we are miles and hours apart.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

A-TSCH	Adaptive Time-slotted Channel Hopping.
ASN	Absolute Slot Number.
BR	Blacklisting Rate.
CB	Cooperative Blacklisting.
CQE	Channel Quality Estimation.
DI	Spectral Disparity.
EH	Spectral Exhaustion.
ES	Exponential Smoothing.
ETX	Expected Transmission Count.
IoT	Internet of Things.
KF	Kalman Filter.
KFES	Exponential Smoothing aided by Kalman Filter.
NFL	Noise Floor Listening.
PP	Path Probability.
SLC	Spectral Link Cost.
TSCH	Time-slotted Channel Hopping.
WSN	Wireless Sensor Networks.

Chapter 1

Overview

Due to advances in technologies of miniaturisation, computing, networking and the reduction in associated costs, the *Internet of Things* (IoT) has become the next stage of development for the Internet [1, 2]. One of the foundations of the IoT is the proliferation of wireless devices which facilitate the provision of pervasive experiences for users, which also leads to an increased demand for limited spectrum resources and poses challenges for IoT devices especially those with low-power wireless sensing requirements. In this thesis we focus on dynamic adaptation of communication based on observed spectral condition in order to mitigate this adverse effect. The context of the research is low-power wireless networks which are susceptible to influences of other more powerful IoT devices. This chapter provides an overview of the research conducted in this thesis. Section 1.1 presents a high-level introduction to the concept and application domains of the *Internet of Things* (IoT). Then in Section 1.2 we motivate our work by identifying the characteristics and related challenges of the IoT. The research questions addressed by this thesis are formally stated in Section 1.3 which also defines the elements of the proposed solution, followed by the contributions of this thesis in Section 1.4. Methods adopted in our research are described in Section 1.5, and assumptions made for this study are summarised in Section 1.6. Finally, an outline of the remainder of the thesis is provided in Section 1.7.

1.1 An Introduction to the Internet of Things

The Internet of Things (IoT) is the emerging paradigm for next generation networks. There have been multiple definitions of IoT due to differing points of view and emphasis placed by different researchers [3, 1, 4, 2, 5, 6], but certain notions are commonly recognised: *Things* refer to objects with virtual identity in the digital domain, whilst the IoT as a whole refers to the worldwide interconnection of those addressable objects based on standard communication protocols. This vision for the future Internet is expected to provide "anytime, any place connectivity for anything" [7] by seamlessly incorporating heterogeneous objects and the existing Internet infrastructure into a cohesive network. Therefore, the concept of IoT can be interpreted as an extension of the existing human-application interaction incorporating the new dimension of *things* [1].

The potentiality of IoT has significant impact on several application domains [1, 4, 8, 9, 10, 11]. In the personal and social sphere, IoT applications can play a role in social media, entertainment and social networking. For example, real-time information about events or places of interest can be collected by objects such as RFID tags and smartphones, and then disseminated on social networks such as Twitter and Facebook [4]. Local news or entertainment information can also be automatically published in forms that fit the capability of specific devices by detecting the presence of nearby smart objects [1].

Another important application domain for the IoT is intelligent environments. In smart residences and office buildings, sensors can be deployed to monitor diverse environmental parameters and humans to support automation. For example, lighting can be adjusted according to sunshine, state of window shutters and user activities [4]. Similarly, the system can automatically tune air-conditioning to ensure the comfort of residents [8]. This application can also be expanded into smart communities such as intelligent college campuses, which may incorporate multi-hop networks between intelligent buildings [9, 10]. This approach

may further improve safety, security and emergency response capabilities on the metropolitan scale.

Healthcare is also a potential beneficiary of IoT technologies. In a hospital environment, smart tags enable the tracking of equipment, specimens, private and confidential materials, which could improve the efficiency of the day-to-day operation and management. Moreover, wearable devices can greatly facilitate healthcare in the community. Patients and particularly those who with limited mobility can be monitored remotely, enabling medical advice to be regularly provided and alert paramedics in case of emergency [4, 8, 11].

IoT technologies are also valuable in industrial settings, for example, the application of RFID and other types of identifiers has already revolutionised modern logistics and retail [4]. The link between product and information enables the "cradle-to-grave" management of individual item lifecycles [1, 8]. In safety-critical industries such as chemical manufacturing, the data made available by *things* provide important information about the status of certain hazardous chemicals in storage, which are essential in safeguarding against potentially disastrous incidents.

1.2 Research Motivation

This vision for the IoT exhibits several characteristics that set specific challenges for its realisation, which establish the starting point of our research. First, the proliferation of wireless devices ensures the ubiquity of intelligent *things* thus paving the way for the emergence of the IoT but also create a critical challenge of localised competition for the wireless medium. To seamlessly integrate with the environment, a vital component of IoT is low-power networks such as Wireless Sensor Networks (WSN) which act as the interface between the physical and digital domains. But the increased volume of objects sharing the limited spectrum places growing pressure on these low-power networks and can result in reduced

opportunities of communication and hence increased risk of performance degradation [12]. The problem of spectral contest is made more notable by the fact that the density of wireless communication in the IoT are not evenly spread but typically concentrates at geographically proximal locations. For example, *things* are present in higher densities at locations where end-users reside or work such as houses and office environments.

Furthermore, implications of the intrinsic unreliability of the wireless medium are not restricted to the access for radio channels at the Medium Access Control (MAC) layer but rather extend to higher layers of the protocol stack. Protocols above the MAC layer that fail to take into account spectrum-related condition could result in inferior performance. For instance, ordinary routing algorithms are typically unaware of the underlying spectrum conditions, a fact which can lead to seemingly good paths whilst they are in fact exposed to undesirable interference.

Finally, IoT systems are expected to operate in a manner which requires minimal human attention. Key to this feature is the ability to autonomously react to stimuli including environmental events or any automated instructions without demanding explicit manual intervention to carry out functions. Consequently, the IoT requires the development of self-adaptive communication technologies.

1.3 Research Questions

Motivated by the challenges identified in Section 1.2, this thesis addresses the question of "what mechanisms can be employed by low-power wireless networks to mitigate the effects of spectral contest on communication performance in typical IoT scenarios?". Specifically:

> (R1) How are low-power wireless networks affected by co-located wireless communication? Can this influence be characterised through measurable performance metrics?

(R2) What mechanisms can be employed by low-power wireless networks to identify interfering nearby communication? How can this detected information be converted to knowledge regarding the desirability of wireless channels?

(R3) Which strategies for adaptation can be adopted by low-power wireless networks to mitigate the effects of interference using the spectral desirability information?

(R4) Can the benefit of adaptation be extended beyond the locality of individual nodes? Can the propagation of this information towards higher layers of the protocol stack help achieve this goal?

In this thesis we introduce a framework for spectrum-aware adaptive communication that can provide answers to the above questions. The following list summarises the key elements of the proposed solution:

(i) The proposed framework should accommodate typical IoT use cases. Specifically, it can support communication for low-power networks in situations of high density of co-located wireless activities.

(ii) The proposed framework should make conservative assumption regarding the hardware capability of the target devices. In particular, single antennas are typical, and processing power and memory are restricted.

(iii) The proposed framework should enable low-power networks to quantify spectral condition in a manner that reflects the level of utilisation of the wireless medium without incurring significant overhead.

(iv) The proposed framework should facilitate dynamic mechanisms of adaptation that allow low-power networks to autonomously adjust their operations on-the-fly.

(v) The proposed framework should support cross-layer utilisation of the information within the framework exploitable by routing protocols.

1.4 Contributions

This thesis proposes a set of adaptive communication techniques which dynamically adjust the operation of low-power wireless devices so that adverse influence of spectral factors can be mitigated. Specifically, the thesis makes the following contributions:

(i) It introduces the Adaptive Time-slotted Channel Hopping (A-TSCH) framework developed to enable low-power wireless networks to adaptively adjust channel access. The design of the framework architecture employs a modular approach. As discussed in Section 3.3, A-TSCH functionality is realised through a series of components, including the detection of co-located communication activities, the formulation of channel desirability using the detection results and a cooperative mechanism to refine the channel hopping sequence.

(ii) It proposes a cooperative blacklisting algorithm in Chapter 6 which is one of the core components of A-TSCH. Blacklisting is a technique which enables nodes to exclude undesirable frequencies from the default channel hopping sequence according to spectral condition. In contrast to the existing global and link-wise strategies discussed in Section 6.2, blacklisting of A-TSCH are conducted through cluster-wise cooperation among multiple network peers as described in Section 6.3. This cooperative blacklisting algorithm yields decisions in a manner which focuses on the performance of overall network rather than individual nodes.

(iii) It presents an investigation into estimation techniques that can be incorporated as methods for learning spectral condition in Chapter 5, which serve as another key component of A-TSCH. A variety of estimators based on Exponential Smoothing and Kalman filter are reviewed and their parameters are tuned through simulation. Estimators in A-TSCH help convert detected ambient interference level to estimated spectral condition upon which blacklisting decisions are generated.

(iv) It provides evidences for the enhancement in performance achieved using A-TSCH through experiments in Chapter 7. Different configurations are considered in the experiments and A-TSCH is assessed in both the control and data planes. Results show that algorithms and modules of A-TSCH behave as expected and resulting transmission success rates exhibit notable improvement compared with those achieved with standard protocols.

(v) It introduces a new metric to RPL routing protocol in Chapter 8 in an exploration of extending spectrum awareness to the network layer. As defined in Section 8.2, the Spectrum Link Cost (SLC) metric is calculated based on blacklisting information within A-TSCH, reflecting the potential risks for data routes from spectral influences and overhead associated with the operation of A-TSCH. As a result, SLC enables more suitable routing decisions to be made for low-power wireless networks.

(vi) It provides supporting evidences for the effectiveness of SLC in both simulation (Section 8.3) and experiments (Section 8.6). Networks of different sizes are modelled and simulations are repeated with random topologies. Experiments are conducted in real-life environment to strengthen our findings in both normal condition and scenarios of sudden deterioration mimicked by using an external jamming device. Results show that RPL using SLC delivers better transmission performance than the default routing metric.

1.5 Research Methods

A review of literature on adaptive communication is carried out in Chapter 2. Surveyed techniques are attributed to different layers of the network protocol stack. Our algorithms and framework proposed in this work are built upon some of these existing techniques, in particular the energy detection (Section 2.2.1.3), clustered architecture (Section 2.3.1.2) and 6LoWPAN (Section 2.4.2).

Measurements of spectral condition and corresponding transmission success rates are taken to form a baseline of communication using standard protocols. Observations are made regarding the influence and characteristics of ambient interference for a selected range of locations.

An explorative approach is adopted in the design of algorithms and framework in this thesis. In Section 5.1, different variants of the Exponential Smoothing technique and Kalman filter are explored as tentative methods for the automatic learning of spectral availability. Moreover, a new estimation algorithm combining the existing methods are described as Algorithm 5.1. Exploration is also made for cooperation strategies of blacklisting. Different from the existing global and link-wise strategies, the four-phased blacklist election (Section 6.3) and selective channel hopping (Section 6.4) are designed taking into account factors including blacklisting accuracy, overhead, scalability and so forth. Furthermore, the introduction of the new SLC routing metric in Section 8.2 represents the explorative attempt of bridging spectral awareness to the network layer.

Simulations are conducted to corroborate the proposed algorithms and metric. Specifically, estimation techniques are implemented in Matlab and their accuracy is compared based on measurements mentioned above (Section 5.3). This approach is also used to examine the effect of blacklisting in Section 6.1. To evaluate the performance of the new routing metric, the Cooja simulator of Contiki is used to model networks with different settings (Section 8.3).

To provide further evidence of the performance of proposed techniques, full implementation of A-TSCH framework and RPL using the SLC metric are developed for a typical platform and experiments are conducted in realistic environments (Section 7.1 - 7.3 and Section 8.4 - 8.6). Evaluation is carried out both in the control plane and related to data transmission performance in the data plane.

1.6 Assumptions

This thesis makes the following assumptions:

- The platforms targeted by this work have restricted capability. For example, in our experiments we use the Guidance and Inertial Navigation Assistant (GINA) [13] and Tmote Sky [14] motes which have maximum transmission power of 3 dBm and 0 dBm, respectively, whilst the typical output power of WiFi devices can be up to 20 dBm [15]. Their memory capacities are limited to 10 kB, and their processors operate at 16 MHz and 8 MHz, respectively.
- Because of their restricted capacity such devices are not expected to be able to hold large numbers of spectral samples for the estimation of channel desirability.
- Devices typically have a single antenna which can carry out signal detection in the license-free Industrial Scientific and Medical (ISM) band or transmit at a given moment.
- Targeted networks consist of statically positioned nodes that operate in environments with significant ambient noise caused by co-located wireless systems utilising ISM band with greater output power.

1.7 Thesis Outline

The remainder of the thesis has the following structure: Chapter 2 reviews related work addressing challenges discussed in Section 1.2 of this chapter. This survey of adaptive communication techniques is organised along the Physical, Medium Access Control and the Network layer of the OSI protocol stack. In Chapter 3, we introduce the Adaptive Time-slotted Channel Hopping (A-TSCH) framework, which is an extension of standard IEEE 802.15.4e protocol and works with 6LoWPAN. The architecture of A-TSCH is described in terms of functional modules,

with specific modules discussed in subsequent chapters. Chapter 4 investigates the relationship between communication reliability and ambient noise using data traces collected in experiments. The main findings from the statistical analysis provide the rationale behind the core modules of A-TSCH framework. Chapter 5 investigates estimation techniques that can be employed for A-TSCH. Specifically, estimators are tuned through simulation and a preliminary comparison between their performance is also provided. In Chapter 6, the algorithm for blacklisting in A-TSCH is presented, which enables the identification and exclusion of undesirable frequencies from channel hopping sequence. The algorithm features a cooperation mechanism for nodes within same clusters, representing a range of trade-offs concerning transmission performance, overhead and scalability. In order to evaluate the performance of A-TSCH, the framework is implemented on a selected hardware platform and tested in a physical environment. Experiments are carried out in Chapter 7 and experimental results are discussed. Then in Chapter 8, the cross-layer enhancement between A-TSCH and the routing protocol is explored by introducing an additional spectrum-related metric. The effect of the proposed change is evaluated through simulation as well as experiments by comparing results obtained with the default and the new routing metrics. Finally, the thesis is concluded in Chapter 9 with a summary and discussions of direction for future research.

Chapter 2

An Overview of Adaptive Communication for the Internet of Things

In this chapter, we survey a range of existing adaptation techniques that can be incorporated into the design of IoT systems to address the requirements and challenges discussed in Chapter 1. Our review follows the reference structure provided in Section 2.1, and techniques discussed in Section 2.2, 2.3 and 2.4 are attributed to the physical, medium access control and network layers of the OSI model [16, 17], respectively. Finally the review is summarised in Section 2.5.

2.1 Adaptive Communication Model

A reference model of adaptive communication for wireless networks in IoT is shown in Figure 2.1, which illustrates adaptation opportunities at different layers of the OSI model [16, 17] and the communication between adaptive wireless networks and IPv6 Internet. It should be noted that the purpose of the model is to highlight points of interest and facilitate the organisation of discussion in the remaining part of this chapter. The model does not assert actual implementation as adaptation can take place at single or multiple layers and techniques may or

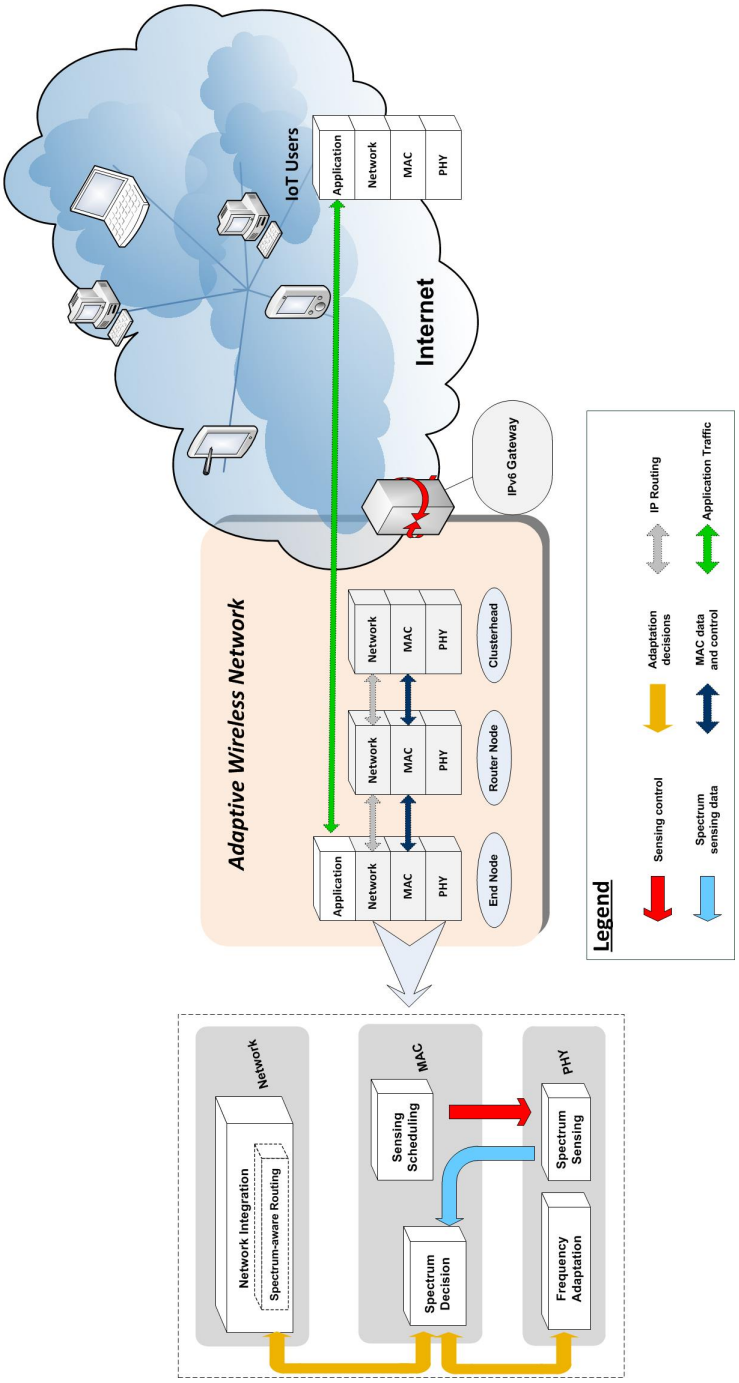


Figure 2.1: Adaptive communication model of IoT. Adaptation opportunities at the PHY, MAC and the Network layer are depicted in the stack to the left. The rest of the figure exemplifies a typical scenario of interactions between existing Internet and adaptive wireless networks through IP. Lightweight adaptation between standard IPv6 and 6LoWPAN is deployed at the gateway.

may not have intra-layer or inter-layer dependency. Furthermore, the terms *End Node*, *Router Node* and *Clusterhead* in Figure 2.1 refer to different roles of nodes and by no means imply disparity in hardware capacity. For example, a Router Node is a device located on the route between an End Node and the destination, and can in the meantime also act as an End Node itself.

As illustrated in Figure 2.1, opportunities and techniques for adaptive communication are mainly identified in physical (PHY), medium access control (MAC) and the network layers. PHY provides the most fundamental infrastructure for any spectrum-based adaptation because of its direct contact with the wireless medium. The importance of PHY is twofold: first, it enables the acquisition of spectral conditions via a group of techniques collectively known as *spectrum sensing* which is essential for well-informed adaptation to be made. Second, the reconfigurability of PHY parameters is indispensable for adapting to spectrum-related events as a great part of adaptation is achieved by altering PHY operating frequency.

The MAC layer controls the behaviours of PHY in adaptation. On the one hand, it regulates and coordinates spectrum sensing activities to ensure spectral information is efficiently collected whilst considering a number of trade-offs. On the other, it generates certain adaptation decisions such as the switching of operating frequency which are in turn put into practice by the reconfigurable PHY.

Two types of adaptation are identified at the network layer. The first one is the incorporation of spectrum information in routing protocols. Routing decisions for wireless networks are usually made based on metrics such as distance or number of hops. Since the spectral condition also has major influences on transmission especially for low-power IoT devices, spectrum-aware routing protocols that take into account spectrum-related events can be adopted to avoid routing decisions that are in conflict with the actual states of wireless medium. This layer also concerns the formation of integrated networks of heterogeneous *things*.

The singular predominance of Internet Protocol (IP) at this layer gives it the potentiality to be established as the "universal language" supported by virtually incalculable types of IoT products [18], eliminating the dependence on one-to-one translation between individual pairs of incompatible technologies and standards. This approach enables *things* to natively communicate with each other using the comprehensively studied IP and greatly simplifies the integration with the existing Internet via lightweight adaptation at gateways [19].

2.2 Physical Layer

The physical layer (PHY) represents the interface between the computational domain and the physical medium. The role of PHY in adaptive communication is twofold. First, it is responsible for capturing spectral information necessary for adaptive processes. Second, it provides the infrastructure for putting adaptation decisions into practise as the selection of operating frequency is realised by configuring corresponding PHY parameters.

2.2.1 Spectrum Sensing Techniques

An important source of interference for wireless communication is caused by co-located devices utilising the same band. Therefore techniques for detecting the condition in certain spectral area, known as *spectrum sensing*, provide one of the foundations of adaptive communication strategies. A number of spectrum sensing techniques have been proposed to deliver such fundamental functionality [20, 21, 22, 23, 24, 25, 26]. Table 2.1 reviews some of the most common spectrum sensing techniques and their characteristics, including prerequisites and limitations that should be considered when choosing the most suitable methods for specific use cases.

Table 2.1: A summary of main spectrum sensing techniques. Matched filter, cyclostationary and eigenvalue-based detection tend to be most suitable where there is a need to distinguish the source of particular target signals. When this is not required for typical IoT scenarios, the energy detection, which has the least prerequisites and incurs the lowest computational overhead, is particularly appropriate for *things* with limited power budget and computational ability.

Technique	**Prior Signal Knowledge**	**Synchronisation**	**Multiple Target Detection**	**Sensing Duration**	**Transmission Disruption**	**Computational Complexity**
Matched Filter [20, 21, 27, 28, 29]	Required	With target signal	With dedicated antennas	Short	None (with dedicated antennae)	Medium
Cyclostationary Feature [21, 30, 23, 26]	Required	Not required	Yes	Long	None (with multiple antennae)	High
Energy Detection [31, 29, 32, 33]	Not required	With adaptive peers	Not supported	Medium	Suspended during sensing periods	Low
Eigenvalue-based Detection [34, 35, 36, 37]	Not required	Not required	Yes	Flexible	None (with multiple antennae)	High

2.2.1.1 Matched Filter

For the detection of known signals, the matched filter technique provides arguably the most efficient and accurate option of spectrum sensing [27, 28, 29]. By identifying and demodulating the target signal, spectrum sensing based on matched filter is able to deliver maximised received signal-to-noise ratio [20] whilst minimizing the effect of other noises. However, the matched filter entirely relies on the availability of *a priori* knowledge of the target signal, which cannot always be guaranteed. Moreover, the complexity of this approach can be considerable, both logically and physically: on one hand, maintaining the coherency with the target signal required for the demodulation introduces additional computational overhead and equipment cost [20]; on the other hand, it also tends to rise if multiple target signals are to be dealt with since individual signals have distinct characteristics and therefore require dedicated receivers [21].

2.2.1.2 Cyclostationary Feature Detection

Another sensing method is cyclostationary feature detection [30]. Similar to the matched filter, *a priori* information about the target signal is required. Since modulated communication signals can be interpreted as multiplexed sinusoidal waves with periodicity, they are classified as cyclostationary [20]. Features of different signals can therefore be identified via spectral correlation analysis [21, 23]. An important advantage of this approach over matched filter is that target waves can be distinguished based on their spectral correlation functions, eliminating the need for multiple antennas where more than one signal is to be monitored [20, 26]. Another favourable outcome of this feature is that regular data transmissions can remain active as signal detections take place since they do not affect one another [26]. However, this method has its limitations. Apart from the reliance on prior signal knowledge, the computational complexity, for example, may be an issue depending on the processing power of the device. Furthermore, identifying signal features could take a relatively long time, consequently adding to the power consumption.

2.2.1.3 Energy Detection

Both of the matched filter and cyclostationary feature detection focus on the accurate and efficient sensing of certain predetermined target signals, built upon the assumption that related *a priori* knowledge is in place. In comparison, energy detection is an alternative spectrum sensing technique which does not require such a precondition.

The concept of energy detection is straightforward: the receiver monitors the signal strength in the operating frequency and the result is compared with certain threshold to determine whether the channel is idle or busy [33]. Although energy detection might not be able to match the performance of matched filter or cyclostationary detection for detecting specific signals [38], it provides a cost-effective option well suited to a wider range of IoT scenarios. In particular, license-free bands are becoming increasingly crowded because of a growing number of wireless applications using these bands such as WiFi and Bluetooth, which results in greater risks of interference. In this case, the focus of spectrum sensing shifts from identification of particular signal sources to channel quality detection for determining the desirable frequencies to use. The study in [32] demonstrates that energy detection is able to deliver satisfactory results with cooperative sensing and sometimes even outperforms approaches based on feature detection. Accordingly, energy detection can serve as an inexpensive yet effective technique and is the most commonly adopted approach for spectrum sensing [39, 31, 32, 40, 26].

2.2.1.4 Eigenvalue-Based Detection

A relatively recent alternative spectrum sensing method is the eigenvalue-based detection [34]. The algorithm first calculates the covariance matrix of the signal samples. The ratio of the maximum to minimum eigenvalue (MME) or energy with minimum eigenvalue (EME) is then computed as the main metrics and its relationship with the detection threshold yields the sensing results [35, 41]. The

detection threshold is key to the performance of this sensing method. The probability of false alarms were employed to deduce the appropriate threshold in a way that its relationship with signal and noise property is eliminated [37].

The eigenvalue-based detection has several advantages over matched filter and cyclostationary feature detection [36]. The eigenvalue-based measure does not require prior signal knowledge or noise patterns, and it is not affected by the noise uncertainty since the metrics and detection threshold is designed to be irrelevant to noise circumstances [36]. The advantages are, however, gained at the expense of large computational overhead [42]. According to [35], the complexity of this method is $M \times L$ times that of energy detection, where M is the oversampling factor or the number of receivers in an multi-antenna scenario and L denotes the sample size.

2.2.2 Frequency Adaptation

Spectrum sensing data obtained at PHY are passed to upper layer processes which formulate environmental awareness and make adaptation decisions [38]. Because adaptation must have a means to be physically realised in order to be meaningful, the ability to adjust PHY parameters on the fly is one of the essential infrastructures for adaptive communications [43, 44]. Because the main challenges this work aims to address are spectrum scarcity and associated interference, the most pertinent PHY parameters is the operating frequency and our work focuses on radio hardware with tuneable operating frequencies.

2.2.3 Summary of Adaptive PHY

The discussion of adaptation at PHY is built on the notion that radio frequency is essentially a scarce natural resource which has been regulated in a static and inflexible manner. As a result, some parts of the spectrum are sparsely utilised whilst others such as the Industrial, Scientific and Medical (ISM) bands are under increasing pressure from more and more IoT devices and technologies operating

in this relatively narrow space. This situation will only become more challenging with the popularity of wireless communication and upcoming new technologies.

The knowledge of spectrum condition provides crucial information for any potential measures aiming to mitigate problems described above. A number of spectrum sensing techniques have accordingly been proposed. The matched filter and cyclostationary feature detector can be used to capture the presence of target signals whose characteristics such as waveform and cyclic patterns are known in advance. Although both provide accurate detection results, the prior availability of information may not be always guaranteed, which limits the scope of their applications. In contrast, spectrum sensing based on energy detection do not rely on such assumptions [35]. Energy detection uses solely the received signal strength to determine the spectral condition. Due to the simplicity, its detection accuracy is subject to noise uncertainty as it is unable to distinguish between multiple signals. Despite this limitation of energy detection, its remarkably low computational complexity and flexibility makes it one of the most popular measures, especially in typical IoT scenarios where spectrum sensing is used for probing the ambient noise level rather than identifying particular signal sources.

By exploiting spectral information obtained at PHY, processes at higher layers of the protocol stack can generate educated decisions on the selection of operating frequencies to minimise the effect of adverse factors and protect communication performance. In summation, PHY techniques discussed in this section are crucial for adaptive communication schemes in maintaining the performance amid growing competition for spectral resources [43].

2.3 MAC Layer

Whilst adaptive PHY makes spectrum-related adaptation possible, it is the media access control (MAC) layer which directly manages the PHY functionalities to

fulfil adaptive missions [16]. On the one hand, it regulates and coordinates spectrum sensing activities to ensure that spectral information is efficiently collected. On the other, it generates adaptation decisions such as switching operating frequency, which are in turn put into practice by the reconfigurable PHY.

Due to the nature of wireless transmissions, signals are subject to a variety of influences such as fading and interference [45] and hence the results of spectrum sensing may not accurately reflect the medium condition. Moreover, the spatial variation of wireless interference could lead to biased channel estimation. Consequently, individual nodes cannot make reliable judgments on spectral condition solely depending on their own local sensing results [26].

The answer to the problem is to form MAC layer cooperation among IoT devices by incorporating spatially distributed entities in spectrum sensing and decision-making procedures [43, 26]. The diversity of their physical surroundings and spectral conditions help reduce the risks of defective or compromised sensing results, leading to improved reliability compared with non-cooperative schemes [46, 47]. In the following, some chief aspects of cooperative spectrum sensing are visited. First, the topological organisation of the cooperation is discussed. And then mechanisms for temporal coordination of sensing activities are reviewed. Subsequently, two types of adaptive decision formulation methods are described and compared. Finally, a summary of cooperative sensing is provided.

2.3.1 Cooperation Architectures

The objective of cooperative spectrum sensing is to combat individual uncertainty or inaccuracy with collective knowledge [48]. Therefore the organisation of cooperation among individual devices has considerable implication on the performance. The architecture for cooperation can be generally categorised as centralised or hierarchical [26] which are reviewed respectively.

2.3.1.1 Centralised Cooperation

The centralised cooperation is the most basic cooperative model, which is built around a singular authoritative entity called fusion centre (FC) [26]. As described in [49, 50, 51, 52, 53, 54, 55, 56, 57], nodes participating in centralised cooperation firstly carry out independent spectrum sensing operation to capture their local spectral condition. Then these local sensing results are reported to the FC through a common report channel. Based on the availability of extensive spectral information from all nodes, the FC generates network-wide decisions which are in turn transmitted to cooperating nodes. Nodes subsequently adjust their communication activities based on instructions received from the FC.

The simplicity of the centralised cooperative model leads to several potential shortcomings. When an increased number of cooperating nodes need to report sensing results to FC, the overhead associated with channel bandwidth and reporting delay tend to grow notably. The computational burden at the FC also surges the amount of reported sensing results increases. Therefore the centralised model is not considered ideal for networks of large scales.

2.3.1.2 Hierarchical Cooperation

The hierarchical model provides an alternative option for cooperation that can alleviate the inefficiency of the centralised model [58, 59, 60, 61, 26, 62, 63, 64, 65]. In the hierarchical cooperative model, nodes are grouped into *clusters* and one of the nodes is elected the *clusterhead*. Depending on specific network structure and setting, clusterheads can be further grouped into clusters of higher level. This clustering process completes when a cluster with the gateway node as the clusterhead is formed.

Cluster members carry out local spectrum sensing and report results to their corresponding clusterhead. A clusterhead generates decisions in a similar manner to the FC in centralised cooperation; but these decisions are only for the particular

cluster it resides. Therefore the hierarchical cooperative model essentially consists of autonomous cooperation within individual clusters. In the following discussion, a number of common clustering strategies are reviewed.

ID-based Clustering Clustering based on node identifiers (ID) is arguably the most straightforward approach for network partition. In this scheme, the role of a node is determined based on its unique identifier [66] which is typically set by administrators or can be extracted from its MAC address. Depending on specific setting, the node with the highest or lowest ID within the communication range is elected the clusterhead. This method has some notable drawbacks in that it does not take into account the node state in clusterhead election. For example, the elected clusterhead may reside at the edge of the cluster and does not have very good connectivity with some nodes at the opposing end. In that case, the ideal clusterhead would have been those in the centre of the cluster. Another potential issue is that the scheme tends to overload a particular set of nodes since the system is inclined to select those with IDs favoured by specific election criteria. Although such a problem can be mitigated by ID rotation, the associated overhead should not be overlooked [67].

Location-based Clustering The location-based clustering algorithms, in contrast, elect the nodes with the most neighbours to be clusterhead [68, 69]. Although the selected clusterheads are arguably the best connected in the cluster, this approach can still cause premature power drainage if some clusters have considerably more members as an imbalanced network load could quickly exhaust the power of busy clusterheads and affect the network lifetime. This approach is also vulnerable to the possible scenario of highly mobile network devices, where frequent updates on clusterhead election are necessary since distances between devices are constantly changing.

Flexible Radius Clustering Clustering technique has been proposed to address the power consumption problem. The approach in [70] constantly updates

the optimal transmission radius of clusterheads using the *Ad Hoc* Network Design Algorithm (ANDA). Cluster members then choose their preferred clusterheads and the resulting cluster arrangement maximises the network uptime before the power of the clusterheads is depleted. However, the study has several limitations, including its dependence on prior knowledge about the locations and quantity of nodes, and the assumption of adjustable transmission power. The most important challenge is that all clusterheads are fixed in advance and the algorithm is only able to update the allocation of cluster members.

Weighted Clustering Weighted clustering techniques elect clusterheads using weight values of individual nodes calculated based on certain criteria. The Weighted Clustering Algorithm (WCA) is proposed in [67]. Weight in WCA combine multiple metrics to improve the battery life of nodes. Specifically, the sum of distances from all neighbours is calculated since distant communication typically tend to be more energy-consuming. Clustering update is only triggered when changes in distances are detected. The mobility of nodes is also considered and less mobile nodes are more likely to be elected clusterheads. Furthermore, to address the potential problem of unbalanced workload for clusterheads, the cumulative duration for which nodes have been clusterheads is used to gauge the extra power consumption.

In WCA, battery life is mainly estimated by the measurement of node uptime. However, it does not consider the fact that nodes may have different startup energy levels and power consumption varies because of communication activities. In comparison, the algorithm proposed in [62] determines clustering weights based on the remaining power level of the node and the number of sent and received packets. This weight represents the capability of acting as the clusterhead and the node with the smallest weight is elected.

Another weighted clustering mechanism based on device energy is proposed as the Energy Efficient Cluster Header Selection (ECS) algorithm in [59]. Clustering

is carried out periodically and the probability for a certain node should act as the clusterhead is computed using its local information such as the initial and current energy, elapsed clustering rounds and how many times it has been elected as the clusterhead.

Spectrum-aware Clustering An adaptive cluster formation algorithm based on spectral environment is devised in [71]. A given node starts listening to all usable channels one by one and: (a) becomes a clusterhead if it hears nothing, or (b) joins an existing cluster as a regular member if it receives a beacon from the clusterhead of that cluster, or (c) goes on to listen in the next available channel if it is more than one hop away from the existing clusterhead.

2.3.2 Spectrum Sensing Scheduling

MAC coordinates not only regular data transmission but also spectrum sensing operation when cooperation is involved [72]. Akin to medium access which can be either random or synchronous, the spectrum sensing can also be scheduled to take place following either of the two approaches [73]. Because of the limitations of the random access approach such as the requirement of dedicated antennae, specific PHY techniques and size of cooperation, many adaptive networks adopt synchronous spectrum sensing which is essential for systems employing energy detection at PHY [26].

The synchronisation for cooperative sensing introduces additional complexities, and one of the common solutions is to adopt Time-Division Multiplexing (TDMA), which divides time into indefinite number of timeslots. Inspired by the TDMA-based technique described in [74, 75, 45], the work in [12, 76] are able to conveniently coordinate the cooperative activities by assigning specific tasks to timeslots. In particular, timeslots can be reserved for noise detection without interference from communications between other nodes.

2.3.3 Cooperative Decision-making

Since the concept of cooperative spectrum sensing is ultimately reflected by the fact that decisions are made by considering information of all cooperative peers, formulating final decisions upon the acquisition of collective knowledge is another critical aspect of MAC in adaptive communication. Depending on the nature of exchanged data, decision-making strategies can be broadly classified as *soft fusion* and *hard combination* [26].

2.3.3.1 Soft Fusion

In networks using soft fusion, cooperative nodes report full spectrum sensing data to the decision-making entities such as clusterheads or FC. The exchanged of raw sensing information could incur drastic overhead because of the theoretically indefinite size of raw data [77]. In order to mitigate this problem, raw spectral data are typically processed before reported for fusion [26]. For example, local spectrum observations can be summarised over a number of collected samples [78, 49] so that the overall amount of exchanged information is reduced. Alternatively, the quantisation of local spectral results can be carried out at the expense of potential loss of precision [77]. For instance, spectrum sensing data are reported in the form of quantised probability density function (PDF) in [56]. Specifically, the distribution of ambient noise strength is computed with respect to a vector of quantisation intervals. The FC then calculate a number of metrics using the PDF vector, which are in turn jointly considered for making adaptation decisions.

2.3.3.2 Hard Combination

With hard combination [79, 48, 80, 54, 81], nodes individually generate their local decisions in the format of binary values which are subsequently reported to decision-making entities [82]. As a result, hard combination scheme is sometimes described as decision counting [83].

Two basic and commonly adopted counting strategies for hard combination

are the OR-rule and AND-rule. The OR-rule [79, 84, 54] generates positive adaptation decision if any of the received local decisions is positive. In comparison, the AND-rule produces positive only if all of local reports agree on a positive decision for adaptation [85]. It is argued in [86] that the choice between the two rules depends on the threshold used by individual nodes. According to its findings, the OR-rule provides good performance if the threshold is considerably high, whilst the AND-rule tends to be optimal in case of a sufficiently low threshold. There is also a number of alternative counting strategies. For instance, analysis presented in [86, 83] suggests that for most of the cases, the optimal counting criterion for yielding positive decisions is that at least half of the location decisions are positive.

Weights can also be applied to local decision counting in hard combination to make refined decisions and several criteria for determining weights are reported in [50]. Device locations [87] and signal-to-noise ratio (SNR) observed by the users [88] are among the basic attributes that can be associated with the weights. Additionally, more complex statistics such as the historical consistency between local sensing results and final decisions [89, 55] can help determine the reliability of certain nodes. Based on these records, local decisions from reputable users are granted greater weight, whilst users that tend to produce inconsistent results are assigned less significance.

2.3.3.3 Hybrid Schemes

Some mechanisms incorporate both the soft and hard mechanisms for optimisation. The scheme proposed in [90] demonstrates one of the examples which uses dual-mode in order to improve signal detection. Hard combination is activated by default with predefined upper and lower thresholds which are shared among all users. The final decision is positive if the result is above the higher threshold and negative if below the lower one. Decisions are then reported to clusterheads or FC. If the detected energy level falls between the two thresholds, nodes do not make local decisions; alternatively, they switch to soft fusion mode and send full local sensing data to clusterheads or FC. Accordingly, the entity responsible

for the fusion process would need to make higher level soft fusion on behalf of indecisive nodes and then combine all decisions to generate the final verdict [90]. A two-stage hybrid data fusion is proposed in [91]. Nodes firstly make local decisions and report them for fusion. If the result is positive, the system perform corresponding adaptation; otherwise, the second stage is invoked where soft data is requested from the adaptive nodes to make more careful decisions [91]. The rationale behind this algorithm is to launch a double-check mechanism using soft data if no consensus is reached using hard combination.

2.3.4 Summary of Cooperative Sensing at MAC Layer

Individual spectrum sensing results obtained at PHY are susceptible to location-specific factors and may not correctly reflect the spectral condition in a network scope. Cooperation in spectrum sensing is therefore considered a suitable solution. This section reviews a range of aspect of the cooperative spectrum sensing at MAC layer.

Two main cooperative architecture are summarised in Table 2.2, revealing a number of advantages of clustered architecture over the centralised version. The most important benefits of clustering are that the decentralisation of decision-making to smaller clusters improves the system scalability and reliability, and that decisions made locally within individual clusters better reflect the specific situations of their physical location. It is worth noting that, according to findings in [95], the probability of path failure tends to increase significantly under potential influences of correlated shadowing which is the fading experienced in multiple paths caused by the same large obstacle such as concrete pillars or trees [95, 96]. As a consequence, this problem can negatively affect cooperation among clusters and hence their overall performance. Accordingly, careful thoughts on node placement are recommended in environments that are prone to correlated shadowing [96]. The clustered structure also reduces the communication overhead. Since reports from cluster members are confined to their groups, the frequency of sending

Table 2.2: Summary of cooperation models. The centralised model treats the entire network as one unitary area, which is simple but not efficient for larger networks. The clustered model, on the other hand, divides the network into multiple manageable clusters where adaptations are carried out autonomously.

Cooperation Model	**Controlling Entity**	**Spectrum Decision**	**Control Channel**	**Traffic Overhead**	**Power Consumption**
Centralised [49, 50, 51, 52, 53, 54, 55, 56, 57]	Pre-appointed Fusion Centre	Network-wide agreement	Unified	High	Inefficient
Hierarchical [68, 69, 67, 92, 32, 93, 94]	Clusterheads	Autonomously made by each cluster	Cluster-wise	Restricted	Mitigated

Table 2.3: Comparison of spectrum sensing scheduling mechanisms. The random access scheme is simple to implement but susceptible to unwanted interference and may require more complex spectrum sensing at PHY. TDMA eliminates the problem of intra-network interference at the expense of maintaining synchronisation.

Scheduling Scheme	**Advantages**	**Disadvantages**
Random Access	Implementation simplicity	Complex PHY techniques required
TDMA	Suitable for simple energy detection	Synchronisation overhead

Table 2.4: Comparison between soft fusion and hard combination. Soft fusion makes use of comprehensive spectrum sensing information and hard combination considers individual binary decisions. Soft fusion may yield more accurate decisions because of the richness of information at the expense of high transmission and computational overhead. Hard combination distributes computational complexity more evenly to every node and may be more suitable for power-constrained devices.

Fusion Mode	**Report Data Type**	**Transmission Overhead**	**Computational Overhead**	**Performance**
Soft Fusion [78, 77, 49, 56, 12]	Detailed sensing data (raw / pre-processed)	High	High	Finer control given the fullness of information
Hard Combination [79, 48, 80, 54, 81]	Binary decision	Low	Low	Subject to limited information of local decisions

and forwarding operation is restricted. Distant transmissions are also minimised as cluster members are usually only one or a reasonably small number of hops away from their clusterhead. These factors together enhance the efficiency and power consumptions, extending the network lifetime. Moreover, the necessity of global communication and report channels is eliminated with clustered sensing network [94]. Instead of using a single controlling frequency throughout the entire network, each cluster is granted the autonomy to individually choose their own channels for exchanging control information and reports [92, 71]. This distributed decision-making mechanism can better accommodate the local spectral situation and greatly improves the network scalability and robustness.

The pros and cons associated with random and synchronous sensing scheduling are generalised in Table 2.3. The scheduling for cooperative sensing is not entirely independent from techniques used at PHY. For example, energy detection is only capable of sampling the strength of signals without distinguishing their characteristics or sources. As a result, network-wide "quiet" periods are required for sensing results to be valid. Synchronous scheduling such as TDMA greatly helps ensure the validity at the expense of synchronisation maintenance. Randomly arranged sensing does not entail such overhead, but options at PHY layer are limited to relatively more complicated techniques such as cyclostationary feature detection. The choice of the scheduling mechanism also influences the effectiveness of adaptation and ultimately depends on specific applications. For example, low-power devices such as wireless sensors usually conduct synchronous spectrum sensing because their limited computational capacity can only support energy detection.

A comparison between soft fusion and hard combination are provided in Table 2.4. Soft fusion centralises the power of decision-making to clusterheads by requiring comprehensive spectral information which can be raw or processed cooperative nodes. With soft fusion, the most comprehensive spectral information

at each node is exposed to the fusion process. This inherent richness of information associated with the raw data enables more sophisticated analysis to be accomplished and thus may lead to a greater chance of more accurate detection and advantageous performance over hard combination [85, 48]. But the overhead in terms of transmission, storage and computation is inevitably high due to the size of exchanged sensing data.

Hard combination distributes the workload of formulating spectrum decisions to all cooperation participants. Nodes individually compute and report their preliminary local yes-or-no binary decisions. Compared with soft fusion, the exchange of binary information reduces pressure on the bandwidth and allows for less complex decision-making algorithms. Although the loss of detail may result in less accurate adaptation decisions [26], it is argued in [48] that hard combination is able to perform nearly as well as soft fusion. Also simulations carried out in [97] show that the performances of soft fusion and hard combination based on energy detection are comparable given a sufficient number of cooperative peers.

2.4 Network Layer

Adaptation at the network layer in the context of IoT mainly concerns challenges of forwarding packets over unreliable wireless links and communication among heterogeneous nodes [98]. Two adaptation opportunities are accordingly identified. The first one is the incorporation of spectral information in routing process. Routing protocols typically measure the desirability of paths based on distance or node states. However, the spectral condition has a major influence on the quality of wireless transmission, especially for low-power IoT devices. Consequently, spectrum-aware routing protocols that take into account spectrum-related events can be adopted to reduce the risks of routing decisions that are in conflict with the actual wireless medium condition. Furthermore, the network layer provides a suitable platform for the integration of heterogeneous technologies and standards operating in IoT due to the predominance of the Internet Protocol (IP).

2.4.1 Spectrum Awareness and Routing

Since wireless links are intrinsically unreliable, routing is usually carried out in an opportunistic way [99, 100, 101], *i.e.*, decisions on next-hops are made at each node by discovering forwarding opportunities [100]. Because the spectrum condition is one of the main defining factors of wireless transmission opportunities, the availability of spectral knowledge provides the routing protocols with additional information and opens up chances of adaptation and optimisation for IoT [102].

Talay and Altilar in [103] propose a link-state routing algorithm aided by spectrum awareness. Each node implementing the scheme maintains its own database of the link cost as with regular link-state algorithms. However, the primary metric for link cost evaluation is channel availability. The spectrum condition is updated periodically and if a link is disconnected because the channel is unavailable, its associated cost grows. In the case that the problem persists, its cost will become so high that it is no longer chosen for packet forwarding. Spectrum Aware Mesh Routing (SAMER) [104] adopts a similar strategy with the difference that the metric used is not just the channel availability but also the quality in terms of bandwidth and error rate. A spectrum utility function is formulated in [105], based on the principle of dynamic back-pressure. Corresponding spectrum utilities are calculated for the set of possible next-hops and the one with the maximum result is chosen.

The preceding examples achieve the optimisation on a per-hop basis, that is, the paths to the next-hop are determined solely based on information at the current node. There has been a number of studies aiming at further enhancement by making per-route rather than per-hop decisions. A relatively straightforward example is presented in [106]. No common control channel is assumed and every node from the source to the one before the destination has to broadcast discovery messages on all channels to determine the next-hops. Nodes also insert local channel availability information into discovery messages so that the destination node learns which channels are usable at every hop and can decide the one to

be used by this particular data flow throughout the route. Despite its simplicity, this algorithm has two main drawbacks. First, the all-channel broadcasting incurs considerable overhead. Although the problem is argued to be alleviated by limiting the number of available channels [106], this would, in turn, increase the vulnerability of the network. Secondly, using a single channel for the entire route means lack of flexibility since it is possible that "good" channels may end up unused simply because they are unavailable between two hops of a lengthy route. Such a problem would add to rather than mitigate inefficient spectrum utilisations, which lead to the need for adaptive communication techniques in the first place.

The mechanism in [107] consists of two phases of operation. Nodes can have more than one entry to the same next-hop on different channels and traffic is randomly distributed among them at every hop. Once adverse spectral events are detected in certain frequency, routing entries based on that channel are disabled. Although this algorithm improves the forwarding reliability by incorporating the spectral diversity and the usage of a common control channel mitigates the broadcasting overhead as in [106], there are still efficiency concerns since the lack of granularity could lead to wasted channel opportunity and overcrowded frequencies in some parts of the network.

The SPEAR routing protocol developed by Sampath *et al.* [108] is based on a similar discovery mechanism as [107] to build a database of channel availability at every node. However, each node broadcasts its local information so eventually the whole map of availability is synchronised within the network. More than one path towards destination can be established by nodes sharing commonly available channels and used for different data flows.

The notion of *route robustness* is introduced in the algorithm devised in [109]. Based on spectral data collected at lower layers, each link between two nodes has a set of probabilities associated with all channels, thus indicating the likelihood

of uninterrupted transmissions in corresponding frequencies. The degree of robustness of a route is then determined by the product of the maximum achievable values of associated probabilities of each link along the way, which is essentially controlled by the most robust available channel. Routes are extended from the source towards the destination via nodes that could form links meeting a certain threshold of robustness [109]. After the elimination of those resulting in loops or leading to a dead-end, the destination node is eventually reached by a set of one of more possible routes. Using an Integer Linear Programming (ILP) model, the final route and channel used in each links are decided. The authors of [109] argue that routes chosen with this method improve the throughput as it exhibits the lowest probability of encountering spectral disruptive events. Simulation results also demonstrate the advantage of adopting this criterion over transmission-rate based route selection strategies [109].

The scheme proposed in [110] demonstrates a different approach by introducing bi-directional activity between the MAC and the network layers. No longer passively accepting the channel assignments, the network layer actively casts effects on the spectrum decision-making in a way that optimises the performance whilst still satisfying lower layer constraints. The work employs the concepts of flow-segment (FS) and maximum flow-segment (MFS) to facilitate channel selection. An FS is defined as a route consisting of consecutive nodes that can communicate using the same frequency. Sometimes there exist multiple FSes from a node towards the destination and the one incorporating maximum number of nodes is the MFS. Transmissions in [110] take place in an on-demand manner. A source node with data to send initiates the dissemination of spectrum information. Starting from the sender, every node captures the local channel availability and forwards a summary of FSes leading to itself in all channels upstream towards the destination node. If at some node X the channel used by MFS from downstream is unavailable, for instance due to interference from other secondary users, node X is then said to be a "decision node" which is responsible for choosing the MFS and assigning the channel for its downstream route. The process

is repeated until the destination is reached and the entire route is established, consisting of the smallest possible number of MFSes, which reduces frequency-switching and therefore achieves lower end-to-end delay and higher throughput. The authors claim the algorithm minimises the amount of channel hopping. It is noted, however, that minimum channel switching is, in fact, only guaranteed for sub-routes between adjacent decision nodes. The resultant channel choice for the entire path is therefore composed of best decisions of sub-routes. However, this scheme does not necessarily yield the optimal overall performance in terms of the amount of channel switching when taking into account the whole map. For example, it is possible that a combination of second-best sub-route decisions may together lead to a better overall frequency hopping reduction.

2.4.2 Network Integration

Another type of adaptation at the network layer is the integration of heterogeneous *things* of different hardware and standards [18] into a unified IoT. The problem of interoperability has been historically addressed by implementing translation mechanisms in the gateway node sitting on the border between the wireless network and the Internet. Gateway nodes are usually connected to a workstation or a wireless access point so as to communicate beyond their local network. By equipping the gateway with translation capability, packets going through the border are examined and reconstructed according to the protocol used by its destined network. A potential problem with this approach is that the wireless network is essentially hidden behind a multi-protocol representative—the gateway, rather than truly integrated into the entire Internet. As a result, it is difficult for a user at the other side of the globe to, for example, access the data of a particular node as if it were a computer, because the header of protocol used may have no field to hold such information. The overlay architecture proposed for opportunistic networks partly mitigates the implementation overhead [98], but the introduction of a new abstraction layer which incorporates various link-level technologies could offset the benefit.

IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [111, 19] have been developed as a more feasible solution to this problem at the network layer. Unlike other layers in the OSI model such as PHY, MAC and transport layer where a variety of standards and technologies operate, the network layer is dominated by Internet Protocol (IP). This renders the network layer an ideal platform for interfacing layers in the integration process [112, 18]. The rationale behind 6LoWPAN is to utilise the well-established and well-understood IP as the universal language commonly used by different technologies, instead of carrying out a time-consuming and error-prone process to conceive of different schemes [111] and later being forced to deal with interoperability issues.

The most critical obstacle to the adoption of IPv6 in IoT is that the transmission of IPv6 headers is too costly for numerous low power devices. 6LoWPAN mitigates this problem by providing a series of header compression mechanisms. For instance, the field of version number can be elided as IP version 6 is assumed to be used. The packet length field can also be inferred from information carried in lower and upper layer payload. The most significant compression is achieved with IPv6 addresses. For both the source and destination address fields, the first 48 bit site prefix field and 16 bit subnet identifier can be elided within the scope of local network and only appended for global use at the gateway. The lower 64 bits of IPv6 addresses can also be inferred from the MAC layer identifier in the best case. Therefore the 40 bytes of fixed IPv6 header can be potentially reduced to as small as two bytes.

Because 6LoWPAN is essentially a compact version of IPv6, only simple adaptation is required at the gateway point to switch between the standard IPv6 and 6LoWPAN packets. The IP nature of 6LoWPAN ensures the entire network is a seamless integration of *things* and the Internet. 6LoWPAN has been employed in a number of studies [113, 12] to form IP-based WSN and the technique is not exclusive to specific applications and can be extended to any power-constrained

IoT devices.

Furthermore, the integration via 6LoWPAN also paves the way for the introduction of IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [114] developed by the Internet Engineering Task Force (IETF). RPL is specifically designed for potentially large scale low-power networks of IoT objects with low or moderate data rates, providing minimised computation complexity and management overhead.

RPL is a distance-vector routing protocol that operate on the Destination Oriented Directed Acyclic Graph (DODAG). A DODAG describe a tree-like network structure where the root is at the gateway node [114, 115]. Consequently, the root is the data sink in typical Multipoint-to-Point (MP2P) applications where sensed data are aggregated at the gateway. Each node in the graph has a *rank* value which indicates its distance to the gateway. The structure of DODAG is accordingly established based on the rank values of individual nodes and data packets follow the path of the lowest rank at each hop until they reach the root. Information required for DODAG maintenance and rank calculation is contained in DAG Information Object (DIO), which is exchanged via a dedicated ICMPv6 packet type called RPL Control Message. By using a trickle timer [116], the exchange of DIO becomes less frequent as the network stabilises, effectively curbing signalling overhead [114].

Since applications in IoT may necessitate distinct routing criteria, RPL has been designed to provide standard architecture that supports a range of different requirements [117]. As detailed in [118], a number of metrics are currently supported to capture the status of node and link. For example, the rank could adopt metric that reflect the historical record of latency or transmission success rates. Constraints can also be used to, for instance, avoid non-encrypted links or neighbours powered by batteries [119].

2.4.3 Summary of Network Layer Adaptation

Adaptation at the network layer extends the performance improvement from local clusters to the scope of network. Spectrum-aware routing protocols can exploit the availability of spectral information at lower layers to determine more appropriate paths across the network to avoid packet loss or optimise frequency utilisation. This layer is also notable for its opportunity for network integration. The predominance of the Internet Protocol (IP) makes it a suitable universal language for miscellaneous IoT product. Although IP is historically considered too heavyweight for low-power devices, IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), a compact version of IPv6, provides a viable solution which fosters rapid implementation. Since 6LoWPAN is intrinsically compatible with IP, heterogeneous low-power IoT devices can communicate efficiently without one-to-one translations and interconnect seamlessly with the vast existing IP-based Internet. The RPL routing protocol also contributes to the prospect of IP-operating wireless networks of low-power smart objects for the IoT.

2.5 Summary

In this chapter, adaptive communication techniques for IoT were surveyed with respect to PHY, MAC and network layer of the OSI model. PHY layer interfaces with the physical environment and is able to capture spectral information via different spectrum sensing techniques. The MAC layer manages the spectrum sensing and adaptation activities at PHY by considering trade-offs between efficiency and adaptation accuracy. The availability of spectral knowledge can also facilitate routing at the network layer and several existing approaches were reviewed. The integration of heterogeneous standards and technologies for IoT can also be addressed at this layer through the universal adoption of IPv6 and its lightweight version 6LoWPAN.

Chapter 3

Adaptive Time-slotted Channel Hopping (A-TSCH) Framework

The previous chapter surveys researches in adaptive communication techniques and identifies a series of opportunities for adaptation in the context of the IoT. Notably, the awareness of wireless medium utilisation is found key to these adaptive measures. This work accordingly focuses on enhancing the performance of low-power wireless networks based on spectral knowledge about medium condition. In this chapter, we propose *Adaptive Time-slotted Channel Hopping* (A-TSCH) framework and present a high-level description of its features and operations. By augmenting the existing channel hopping mechanism of IEEE 802.15.4e [120] with the capacity for learning spectral condition and cooperative blacklisting, A-TSCH provides spectrum-aware adaptive communication for low-power wireless networks.

In Section 3.1, the existing channel hopping mechanism of IEEE 802.15.4e is reviewed. Then the the design rationale and key features of A-TSCH are introduced in Section 3.2. Section 3.3 provides a high-level framework architecture of A-TSCH, where operational cycles are described and important components are identified for detailed discussions in subsequent chapters. Finally, summarising remarks are provided in Section 3.4.

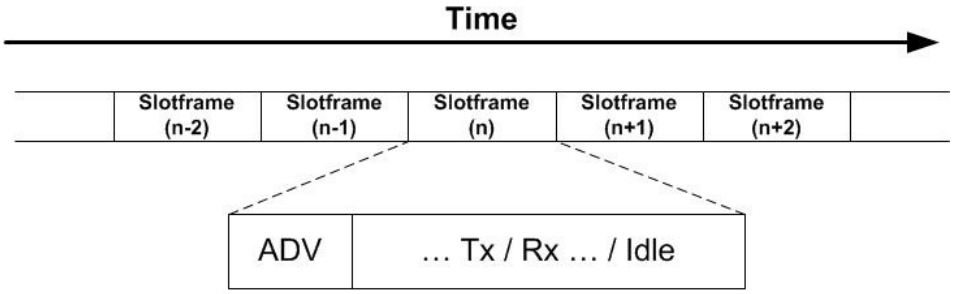


Figure 3.1: Slotframes of standard TSCH include ADV, Tx and Rx slot types.

3.1 Standard IEEE 802.15.4e TSCH

IEEE 802.15.4e is an amendment to IEEE 802.15.4 standard which aims to increases robustness against external interference and multi-path fading [120]. It addresses these problems through the inclusion of Time-slotted Channel Hopping (TSCH) technique. TSCH divides time into a series of infinite successive *timeslots*. Individual timeslots are assigned slot type values which define the operation to take place within their durations. There are by default four slot types in standard TSCH: Advertisement (ADV), Transmission (Tx), Reception (Rx) and Idle. Behaviours of network nodes are governed by the operation scheduled for the current timeslot at a given moment. A group of timeslots forming a pattern of slot types repetitive along the time axis constitutes a *slotframe*, as illustrated in Figure 3.1. Every network node can be seen as iterating operations shaped by its slotframe, although the content of the slotframe itself may be adjusted. As a consequence, content of slotframes of individual nodes need be scheduled in a coordinated manner and nodes must by time-synchronised for the network to function properly.

The count of total elapsed timeslots since the start of the network is recorded as *Absolute Slot Number* (ASN) in TSCH to achieve a network-wide common sense of time. ASN is disseminated through advertisement messages exchanged

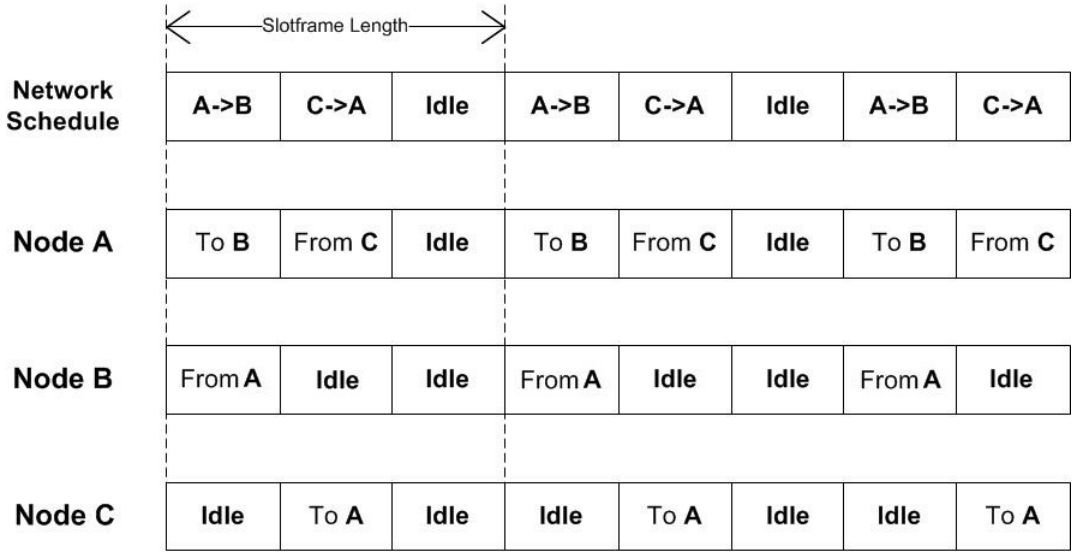


Figure 3.2: Example schedule with slotframe length of 3. Node A, B and C share the same slotframe structure and perform in a coordinated manner.

in *ADV* slots. Nodes use ASN to deduce their temporal position within the slotframe which ensures that they simultaneously enter and exit the same timeslots and operate according to timeslot types defined by their individual schedules. A simplistic example of slotframe schedule with three slots for a network of three nodes running TSCH is illustrated in Figure 3.2. Node A is set to transmit to node B in the first timeslot. Therefore this timeslot is assigned to the communication between A and B from the perspective of the network. The other slots are also scheduled the same way, and operations repeat every three timeslots. Scheduling mechanism of TSCH is beyond the scope of this thesis and is unmodified in A-TSCH.

$$\text{Channel} = (\text{ASN} + \text{channelOffset}) mod\ 16 + 11 \tag{3.1}$$

In addition to the deduction of slot types and associated operation, ASN is also used to determine the operating frequencies for channel hopping. Based on

Equation (3.1), channel numbers between 11 and 26 are yielded for positive integer ASN values, hence providing indiscriminate pseudo-random access to all sixteen IEEE 802.15.4 channels in 2.4 GHz band. The additional factor of *channelOffset* can be optionally set to yield different channel hopping sequences.

In summary, the standard TSCH allows nodes to carry out their individually scheduled communication in a coordinated manner using pseudo-randomly chosen channels. This infrastructure is inherited in the design of A-TSCH in this thesis.

3.2 Overview of A-TSCH

3.2.1 Design Rationale

Although the standard IEEE 802.15.4e TSCH is able to enhance the communication reliability by evenly spreading negative effects of adverse factors over the entire set of available IEEE 802.15.4 frequencies [45], its indiscriminate utilisation of channels poses an important limitation. As degrading factors such as fading and interference are generally not uniform across the spectral space, some frequencies are affected to greater extent than others and hence exhibit less desirable quality for wireless transmission. Communication using standard TSCH inevitably takes place in those less desirable frequencies at some point, which potentially compromises the overall performance (experimental evidence of such effect is provided in Chapter 4). A-TSCH is designed with the aim to mitigate this problem.

3.2.2 Key Features

The Adaptive Time-slotted Channel Hopping (A-TSCH) is designed to improve the performance of the standard TSCH whilst preserving its benefit. Recognising the diversity of channel desirability, A-TSCH identifies and excludes channels of undesirable quality from the channel hopping sequence. As a result, the network constantly switch operating channels only within the subset of desirable ones.

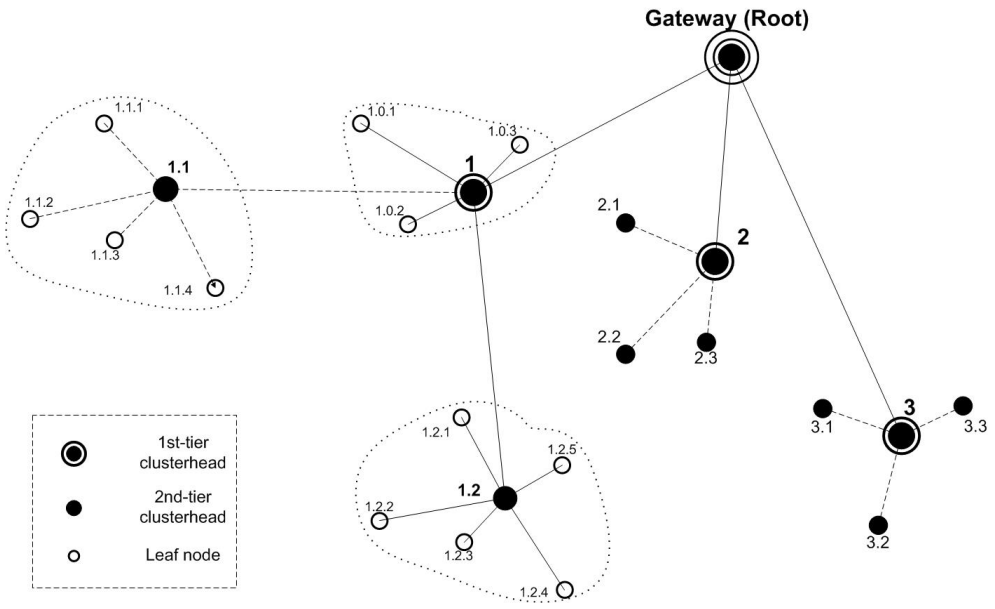


Figure 3.3: A-TSCH assumes tree-like topology with hierarchical cluster structures.

The collection of undesirable channels is known as *blacklist* and the process of generating and applying a blacklist is referred to as *blacklisting*. Key features of A-TSCH include:

- Detection of spectral condition through ambient noise floors.
- Learning of channel desirability based on spectral condition records.
- Cooperative blacklisting
  - Generation of blacklists through cluster-wise cooperation.
  - Selective channel hopping using blacklists.

The A-TSCH framework operates with tree-like network topology rooted at the gateway node such as the one illustrated in Figure 3.3. A-TSCH also considers such topology a hierarchical structure of clusters. A cluster is a subset of nodes grouped based on certain criteria such as physical proximity [121]. Every cluster has one appointed *clusterhead* which forwards data from and to other

cluster members. Nodes that do not forward data from others are *leaf nodes*. For example, nodes *1*, *2* and *3* are first-tier clusterheads directly connected to the gateway in Figure 3.3, whilst *1.1*, *1.2* are second-tier clusterheads. In a typical Multipoint-to-Point scenario where data travel from leaf nodes to the gateway (data sink), clusterhead node *1.1* aggregates data from its cluster members nodes *1.1.1*, *1.1.2* and *1.1.3*, and then forwards them to the corresponding first level clusterhead *1* before reaching the destination.

3.3 A-TSCH Operation

A high-level flowchart of A-TSCH operations is provided in Figure 3.4. Because A-TSCH operates on the basis of slotframe, logical paths in the figure represent procedures of the operational cycle in a given timeslot. Square boxes denote important A-TSCH functional modules. Control flows are directed by arrowed lines annotated with specific actions. Two separate logical branches are explicitly marked by hollow and filled arrow shapes, respectively. In the remainder of this section, we go through different stages of the A-TSCH operation to demonstrate modules and their interactions. In particular, two most complex modules are identified and detailed subsequently in Chapter 5 and 6.

3.3.1 Slotframe Scheduling

The schedule of slotframe is checked at the very first stage of every A-TSCH cycle to determine the slot type and associated operation. To facilitate the detection of spectral condition, A-TSCH introduces an additional *Noise Floor NF* slot type to the slotframe. Nodes suspend data transmission and probe the strength of ambient noise in NF slots. This arrangement is necessitated by the limitation that energy detection technique is not capable of distinguishing own communication signals and ambient noise, as discussed in Chapter 2. An example A-TSCH slotframe is illustrated in Figure 3.5 with two NF slots placed at the rear.

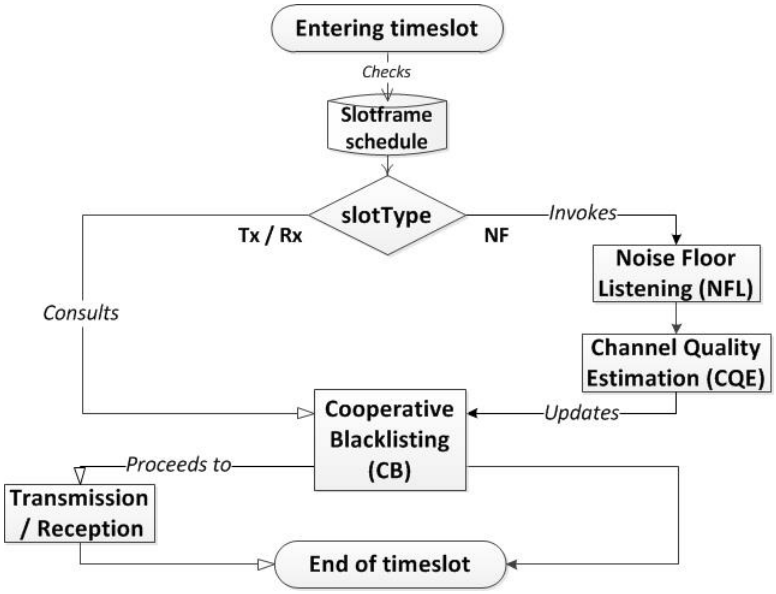


Figure 3.4: Diagram of A-TSCH logical cycles. Arrowed lines are annotated with actions and the two arrow types correspond to different control flow branches.

The slotframe schedule directs A-TSCH operation into corresponding logical branches. In Figure 3.4, operation in NF slots is directed by filled arrows. The other branch is directed by hollow arrows, representing transmission and reception of data in Tx or Rx timeslots. Since this new slot type is unknown to existing IEEE 802.15.4e, nodes running standard TSCH ignore NF slots and treat it as Idle type.

3.3.2 NF Cycle

Upon entering an NF slot, the ASN number is input to Equation (3.1) to pseudo-randomly yield the channel whose noise floor is to be sensed. The *Noise Floor Listening* (NFL) module retrieves the energy level of channel noise from the Received Signal Strength Indicator (RSSI) register of the radio chipset [12]. RSSI is chosen as the primary indicator of spectral condition as it is found to exhibit reasonable correlation with transmission success rates in several studies [122, 123, 124], and a detailed investigation into the effectiveness of this measure is presented in Chapter 4.

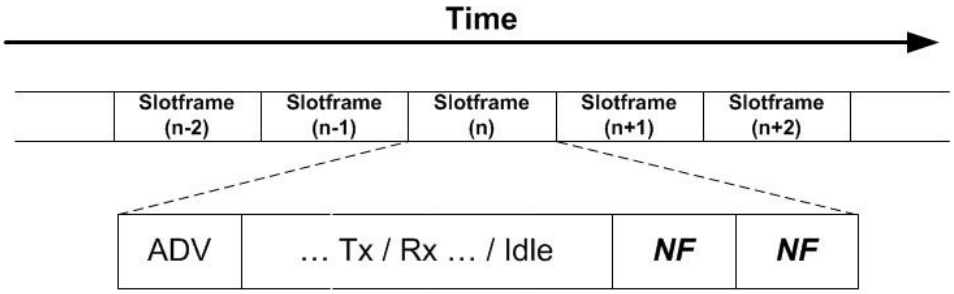


Figure 3.5: A sample slotframe used in A-TSCH. Compared with standard TSCH, an additional NF slot type is introduced.

Since sampled ambient noise levels are typically subject to momentary fluctuations of ambient energy strength, individual noise floor readings may not be sufficient to accurately capture the characteristics of certain frequencies over longer periods. To mitigate this issue, the sampled noise floors are subsequently passed to the *Channel Quality Estimation* (CQE) module. Employing certain estimation technique, the raw noise readings are processed and transformed into estimates of channel noise floors which indicate the quality of individual channels for wireless communication. The CQE module provides an important component of A-TSCH and is described in detail in Chapter 5.

In the next stage, the results of CQE are passed to the *Cooperative Blacklisting* (CB) module which is detailed in Chapter 6. The CB module is responsible for identifying subsets of undesirable channels known as *blacklists*. The module firstly triggers nodes running an A-TSCH instance to independently identify their own candidates of blacklist by drawing on their local channel quality estimates. Then based on the hierarchical network model previously described, an agreement on the blacklist to be imposed on communication within this cluster is reached through cooperation among proximate cluster members. In addition to the cooperative generation of blacklist which is a crucial function of A-TSCH, the CB module is also responsible for the exertion of blacklists in transmission

and reception, which is subsequently described in cycles for Tx and Rx slots.

3.3.3 Tx / Rx Cycle

Similar to the NF cycle, an operating frequency is pseudo-randomly generated using Equation (3.1) at the commencement of Tx or Rx timeslots. Then the module of CB is invoked to find out whether this tentative operating channel is included in the blacklist yielded in NF slots. If the channel is not blacklisted, nodes carry on with transmission and reception as normal; otherwise, a new operating frequency must be generated and again checked for validity. The process iterates until a usable channel is yielded. This practice effectively makes the A-TSCH hopping sequence selective about the channel quality detected in NF timeslots.

The selective channel hopping is another core function of the CB module. Details are provided in Section 6.4, describing the adaptive process for deciding pseudo-random channels which considers trade-offs between blacklisting accuracy and overhead.

3.4 Summary

This Chapter provides a high level description of A-TSCH framework including its core modules and their logical cycles. To enhance the indiscriminate usage of frequencies of the standard IEEE 802.15.4e TSCH, A-TSCH makes channel hopping selective about the quality of channels, measured in terms of ambient noise floor levels. Additional NF slots are specifically introduced to facilitate the detection of noise strength. The Received Signal Strength Indicator (RSSI) is retrieved from the radio chipset when normal data transmission is suspended in NF slots. Noise readings are processed by CQE module to form estimates of channel quality. In the subsequent stage of Cooperating Blacklisting, nodes individually propose candidate channels to be blacklisted, and then decisions on blacklist for each cluster are made via cooperation among cluster members.

Pseudo-randomly generated channels for communication in Tx / Rx slots must be validated before transmission can take place. Channels found in the blacklist must be replaced by an admissible alternative channel. In the following chapters, properties of ambient noise as the primary spectral condition indicator is first investigated in Chapter 4. And key modules of *Channel Quality Estimation* and *Cooperative Blacklisting* are discussed in detail in Chapter 5 and 6, respectively.

Chapter 4

Ambient Noise Floor and Channel Desirability

As described in Chapter 3, A-TSCH detects the desirability of specific IEEE 802.15.4 channels by probing their noise floors. In this chapter, the properties and effectiveness of this spectral indicator are investigated. To facilitate our investigation, communication traces and ambient noise samples are collected using different configurations at multiple locations. In Section 4.1, environments and tools used for data collection are introduced. In Section 4.2, the relationship between ambient noise and communication performance is observed. In Section 4.3, measurement of ambient noise is conducted over prolonged periods of time and features of collected samples are analysed. Finally this chapter is summarised in Section 4.4.

4.1 Experimental Setting

This section introduces the environments used for the collection of communication traces and monitor of ambient noise. The specification of hardware and software used for experiments is provided.

4.1.1 Environment Information

Because spectral conditions usually vary among environments, experiments were conducted at different locations to reflect diverse physical surroundings and radio medium utilisation. Experiment locations are identified as A, B and C. Location A is an open workspace with heavy presence of WiFi signals. Three wireless routers are deployed on-site and more than 10 WLANs can be detected on normal weekdays. Location B is a flat room in a residential building. Only one wireless router is installed in the property and typically less than three other WLANs are visible at weaker signal strengths. Location C is a departmental office in a college building block. Only one WLAN connection is steadily detectable at Location C, and Internet access in the block is predominantly via cable sockets. Key facts of these locations are summarised in Table 4.1.

4.1.2 Device Specification

Guidance and Inertial Navigation Assistant (GINA) motes [13] are the primary hardware platform used in this chapter and the remaining part of the thesis. GINA is designed by the Kris Pister group at the University of California at Berkeley. GINA comes in different configurations and the type of motes used in this work is shown in Figure 4.1 and specified in Table 4.2. Since the focus of this work is on the communication in IoT, on-board sensing units have been greatly reduced for our GINA motes. Instead, they are equipped with UART-to-USB bridge chips and mini-B USB connectors to facilitate the retrieval of experimental data.

IAR Embedded Workbench [125] has been used for developing the implementation on GINA motes. IAR provides a tool chain for the compilation of code and debugging motes through Joint Test Action Group (JTAG) adapters.

Table 4.1: Profile summary of environments used for experiments. The environments are identified as Location A, B and C, and are hereafter correspondingly referred to by their ID.

Location ID	**Profile**
A	- Communal workspace - Approximately $12m \times 12m$ - Three wireless access points placed in proximity - Typically more than 10 WLANs visible
B	- Residential flat room - Approximately $4m \times 3m$ - One wireless access point installed in the property - Typically about 3 WLANs visible
C	- Departmental office room. - Approximately $5m \times 2m$ - No wireless access point installed inside the room. - Typically one WLAN connection steadily detectable

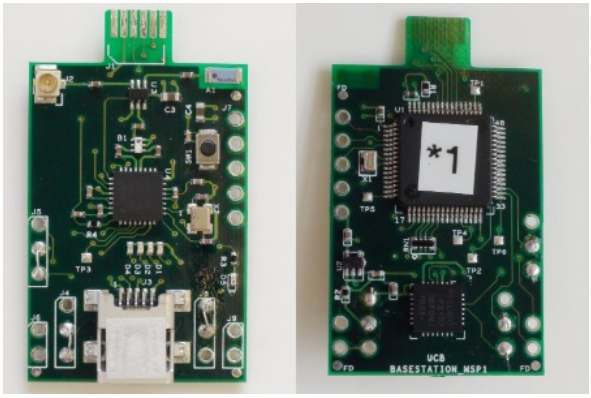


Figure 4.1: Guidance and Inertial Navigation Assistant (GINA)

Table 4.2: Specification of Guidance and Inertial Navigation Assistant (GINA) motes [13].

Type	Manufacturer	Part	Details
Microcontroller	Texas Instruments	MSP430F2618	16-bit, 16MHz, 116kB flash, 8kB RAM
Radio	Atmel	AT86RF231	IEEE802.15.4-compliant, 2Mbps-capable
Antenna	Rainsun	AN3216-245	2.4GHz
USB connector	Hirose	UX60A-MB-5ST	Type B, receptacle, female contacts, surface mount, right angle
UART-to-USB bridge	Silicon Labs	CP2102	28-pin, slave USB interface, QFN-28
JTAG port	N/A	N/A	connects to the GINA-JTAG-adapter

4.2 Communication under Ambient Noise

This section focuses on wireless communication under the influence of ambient noise. First, the experimental configuration for data collection is provided. Then a number of metrics are introduced to facilitate our analysis. Subsequently, observations are made on channel noise and its impact on low-power wireless communication.

4.2.1 Configuration for Data Collection

Two GINA motes configured as a pair of sender and receiver were deployed in Location A, B and C, respectively. In each run of experiments, the sender was programmed to transmit batches of 100 packets with inter-packet time of one second using one of the IEEE 802.15.4 channels. The receiver was made to constantly listen for packets in the same channel. Also operated in the receiver was a sniffer program which kept probing the ambient noise floor of the operating channel in the intervals between the arrival of packets. Here the noise floor samples specifically refer to energy level readings retrieved from the Received Signal Strength Indicator (RSSI) register of the radio tranceiver, which reflect the ambient signal strength in the operating channel. These values were not affected by the pair's own signal because readings were obtained during intervals between packet transmission and reception. Successfully received packets and noise readings obtained in intervals were all forwarded to serial ports and recorded. Once the transmission of a batch was finished, the sender and receiver synchronously switched to a different IEEE 802.15.4 channel and repeated the process for the next experiment run. Accordingly each run took approximately 100 seconds and collecting experimental data of all IEEE 802.15.4 channels in a round-robin fashion took 1600 seconds. Each mote pair continuously collected data for 24 hours, therefore approximately 54 rounds were carried out at each location.

4.2.2 Metrics for Analysis

Two categories of metric are devised to extract these information from the collected traces for analysis in this section.

The first category is Expected Transmission Count (ETX) which quantifies the transmission success rates. As defined in Equation (4.1), ETX is the reciprocal of Packet Delivery Ratio (PDR), the percentage of successful delivery in all transmission. Accordingly ETX denotes the average number of transmission attempts required for a packet to be successfully received [126, 45]. ETX takes the range between $[1, +\infty)$ where ETX of 1 indicates perfect transmission with no lost packet.

$$ETX = \frac{1}{PDR} = \frac{Packets_{transmitted}}{Packets_{received}} \tag{4.1}$$

For analysis in this section, ETX is calculated for each packet batch, denoted by ETX_{batch}. Because individual batches were transmitted using a single channel in its entirety, each ETX_{batch} represents the transmission success rates of one of the IEEE 802.15.4 channels. Consequently, the overall performance of individual channels can be measured by grouping and averaging corresponding ETX_{batch}, and this resulting metric is defined as $ETX_{channel}$.

The second metric category of is *Noise Floor Strength* (NFS), which we define as the level of ambient noise that the transmission of individual packets is exposed to. In terms of data collected for this section, NFS are calculated by the receiver as the mean RSSI within the intervals between the reception of packets. Accordingly, for each received packet there is a corresponding NFS that characterises the operating channel used for its transmission. Similar to ETX metrics, NFS_{batch} and $NFS_{channel}$ are also introduced. NFS_{batch} is computed as the mean NFS

of specific packet batches. $NFS_{channel}$ is the average NFS_{batch} of each IEEE 802.15.4 channels.

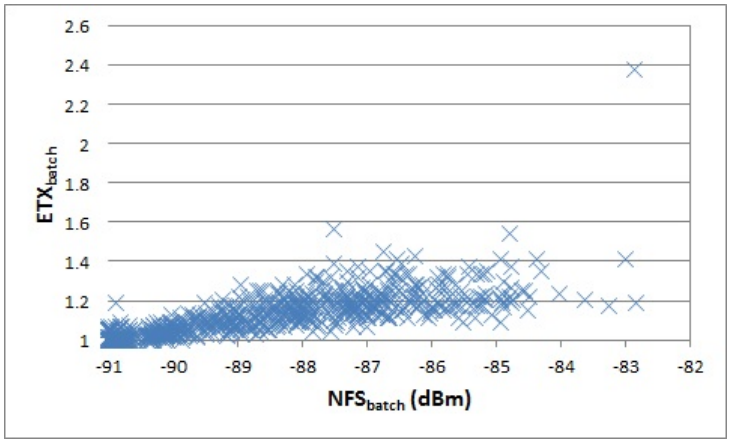
4.2.3 The NFS-ETX Relationship

The general relationship between ambient noise and transmission can be observed from (NFS_{batch},ETX_{batch}) pairs plotted in Figure 4.2 for three locations, respectively. In Figure 4.2a for Location A, it is observed that the ETX are predominantly lower than 1.1 when the noise floor is less than approximately -90 dBm. As the noise floor grows from -90 dBm, ETX grows towards 1.4. It is also noticed that ETX tends to show increased variation ranging between 1.1 and 1.6. Although the NFS-ETX correlation does not appear strictly linear, the negative effect of ambient noise floor on ETX performance in terms of both magnitude and uncertainty is apparent.

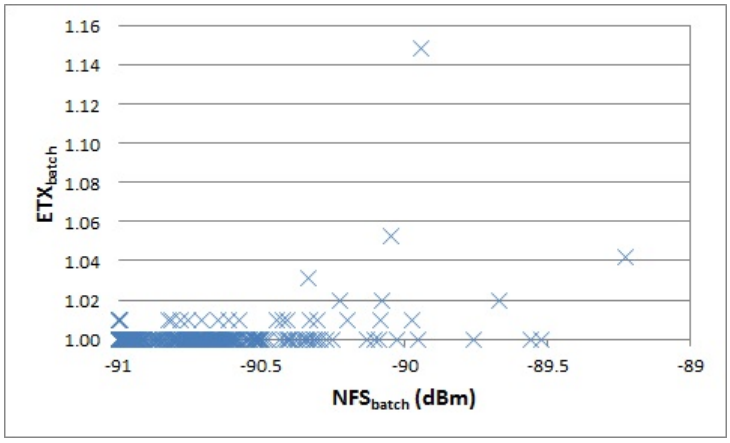
The NFS-ETX relationship at Location B illustrated in Figure 4.2b is in agreement of the observation about Location A. When NFS_{batch} is below approximately -90.7 dBm, the vast majority of ETX have the value of 1, meaning all 100 packets were successfully received. Above this noise level the ETX tend to dispersed away from 1 and fluctuate in greater scales. Compared with Location A, the ETX distribution is more concentrated towards 1 and overall tendency appears less regular.

For Location C, the (NFS_{batch},ETX_{batch}) pairs in Figure 4.2c also exhibit the trend of growing magnitude and instability of ETX as NFS increases. In particular, the value of ETX becomes notably higher and variable once the NFS is higher than approximately -89 dBm. Although there is a lack of linearity compared with features observed in Figure 4.2a, findings in data of Location C are still in agreement of the general tendency.

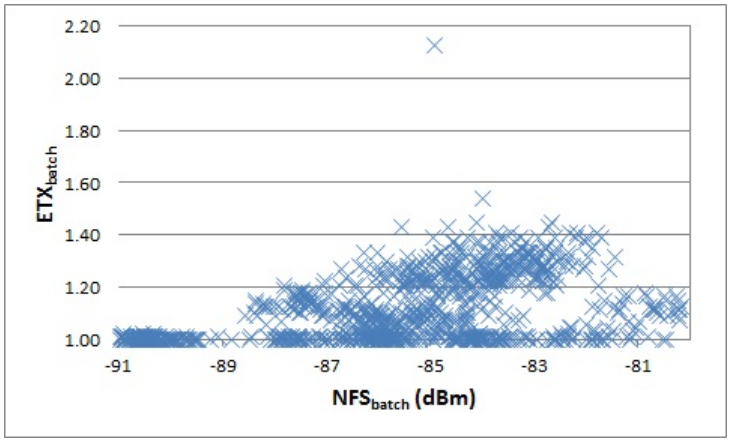
Based on the above findings, the following observation is reported:



(a) Location A



(b) Location B



(c) Location C

Figure 4.2: Relationship between noise floor and transmission success rates. Horizontal axis represents NFS_{batch} and vertical axis denotes the corresponding ETX

Observation 1 *The performance of transmission is negatively affected by the ambient noise floor of the utilised IEEE 802.15.4 channel. As the noise floor increases, the reliability and stability of transmission tends to decrease.*

4.2.4 Channel Diversity

The $NFS_{channel}$ and $ETX_{channel}$ of individual IEEE 802.15.4 channels are plotted in Figure 4.3 for Location A, B and C, respectively. As shown in Figure 4.3a, noise floors vary visibly between channels at Location A. In particular, channel (15 - 20, 25 - 26) have very low $NFS_{channel}$ and channel (12 - 13, 22 - 23) are notably noisy. Meanwhile, variation among $ETX_{channel}$ shown in Figure 4.3b exhibit a pattern in accord with the difference in $NFS_{channel}$. Specifically, ETX are over 1.12 for channels with $NFS_{channel}$ at approximately -87 dBm, and mostly below 1.04 for channels with $NFS_{channel}$ less than -90 dBm.

Similar features are also shown in $ETX_{channel}$ and $NFS_{channel}$ for Location B. In Figure 4.3c and 4.3d, the patterns of low NFS - low ETX and high NFS - high ETX are present. Specifically, ETX reaches the highest level in channel 18 and 19 wher the average noise floors are found to be the strongest.

For Location C, results in Figure 4.3e and 4.3f again show the similar correspondence of ETX with NFS. Channel (11, 16 - 17, 21 - 24) exhibit the most notable noise floor strength and are associated with the highest ETX values. Channel (13 - 14), in comparison, have the lowest NFS and the best ETX performance.

Despite the different noise conditions at experimental locations, the ETX achieved in individual channels are invariably found to be related to the corresponding noise floors. Furthermore, there is a variation in patterns of channel noise floors in the 2.4 GHz band shown Figure 4.3 which appears to be environment-specific as no static pattern can be identified. Consequently we

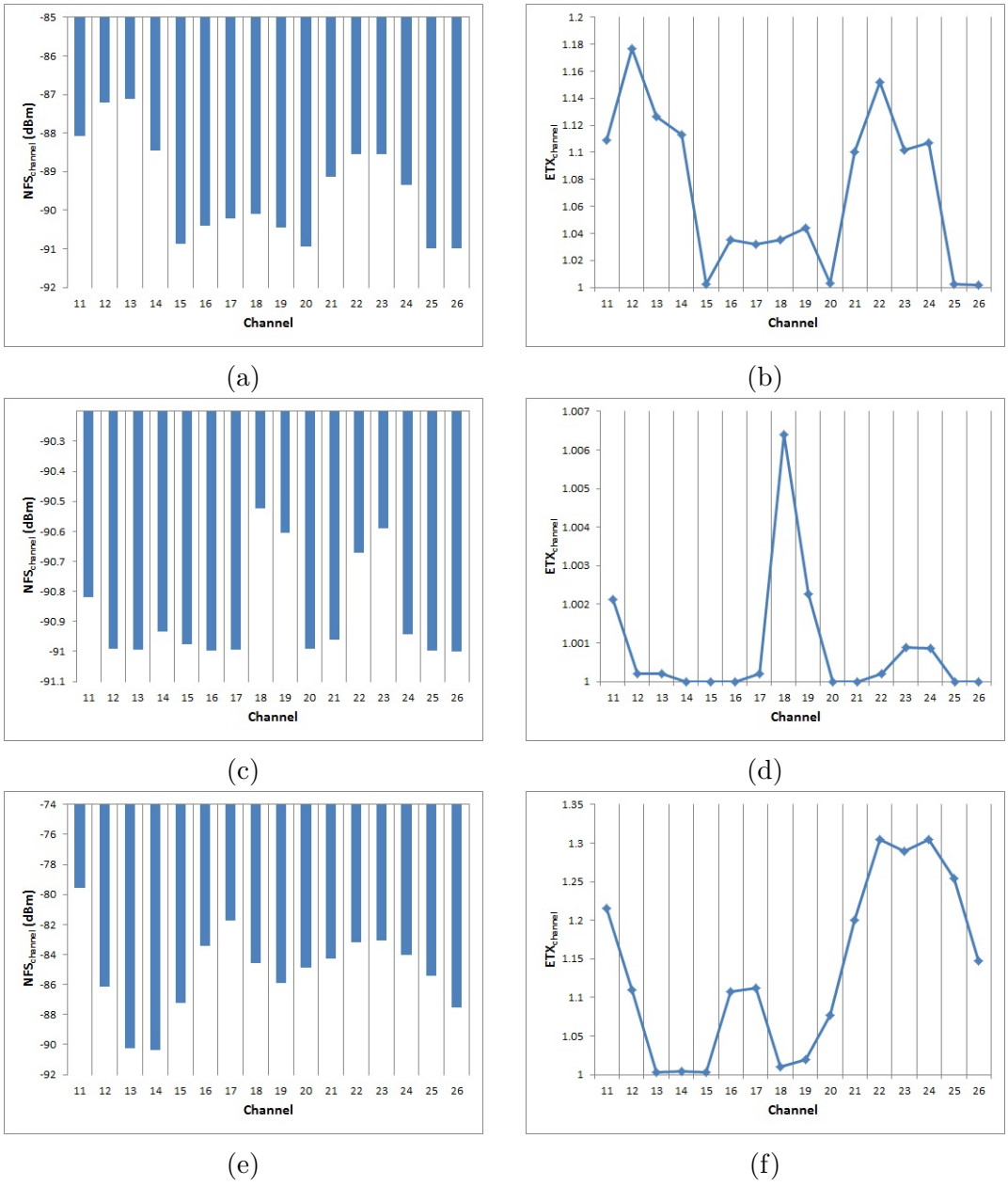


Figure 4.3: Noise floor and transmission success rates of individual channels at different locations. The left column depicts $NFS_{channel}$ and the right column illustrates $ETX_{channel}$. (a) and (b), (c) and (d), (e) and (f) are for Location A, B and C, respectively.

report the following observation:

Observation 2 *The difference of achievable transmission success rates in individual IEEE 802.15.4 channels is in accord with the environment-specific diversity of their ambient noise floors.*

4.2.5 Noise Threshold for Transmission Success Rates

Observation 1 and 2 have demonstrated the negative effect of ambient noise on transmission and its correlation with unequal performance of IEEE 802.15.4 channels. In the next step, we examine whether there exist certain noise floor levels of special significance in defining the transmission performance.

It is noticeable in Figure 4.2 that the correlation between NFS_{batch} and ETX_{batch} is much more deterministic at Location A and C than at Location B. The clear rising trends in ETX with increasing NFS observed in both Figure 4.2a and 4.2c is lacking in Figure 4.2b. A closer look shows that this difference does not create contradiction but reveals some insights into the effect of ambient noise. Taking into account the absolute scale of the ETX_{batch} axis, the entire plot area of Figure 4.2b can only account for a confined part in Figure 4.2a or 4.2c where the overall noise levels are very low. In other words, traces collect at Location B only captures ETX under very weak ambient noise. The relative irregularity present in Figure 4.2b indicates that the influence of ambient noise on wireless transmission is not always significance but subject to specific strength, leading to the following observation:

Observation 3 *There is certain ambient noise floor threshold above which communication is significantly affected.*

To identify suitable noise threshold, (NFS_{batch},ETX_{batch}) records are statistically investigated in Matlab using two different approaches. We first employ the clustering technique which is used for dividing data into smaller groups based on certain criteria so that data belonging to the same cluster exhibit homogeneous

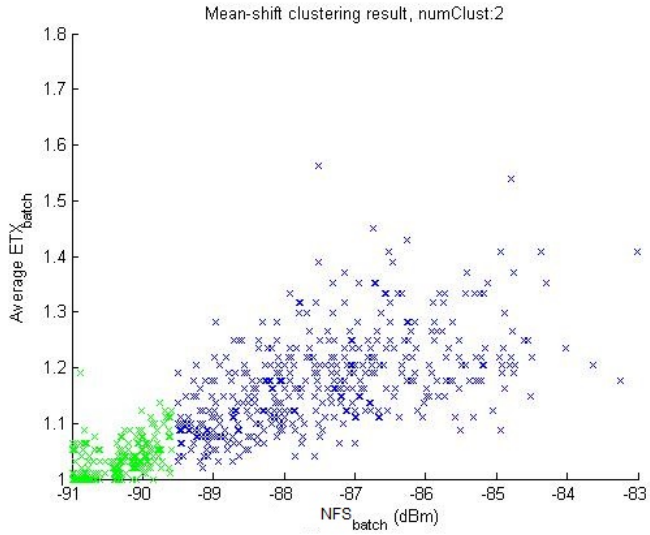


Figure 4.4: Clustered NFS-ETX records

characteristics whilst those in different clusters are as distinct as possible. In this work, the mean-shift clustering technique is adopted. Mean-shift is essentially a mode-finding technique: it treats the data space as empirical probability density function and locates the value which maximises its probability mass function, known as the mode. Therefore mean-shift clustering is able to recognise local maxima of the density and clusters associated with them [127]. Because one characteristic that distinguishes noise areas (clusters) separated by the threshold is the noticeable variation in NFS-ETX correlation, we anticipated that two clusters of distinct levels of density could be identified by the mean-shift technique which correspond to regions where ambient noise has dominant or relatively less significant influence on wireless transmission.

As depicted in Figure 4.4, two clusters are identified which is in agreement with our earlier anticipation. The difference in distribution density of ETX with respect to NFS represents the different extent of effect that ambient noise exerts. The boundary between these two clusters lies between -90 and -89 dBm. Because low-power wireless devices focused in this work are relatively inefficient in

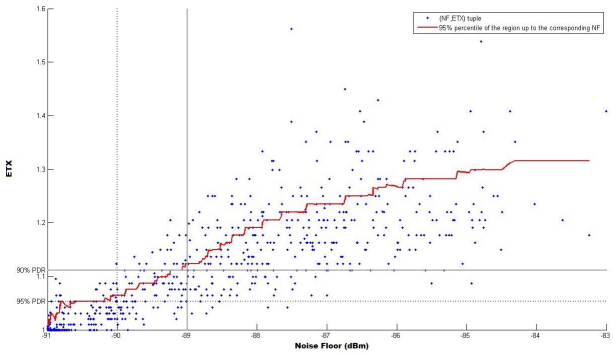


Figure 4.5: Records of NFS-ETX with 95th percentile reference shown in the red line. The 95th-percentile reference separates 5% of the population and the rest 95% based on ETX values. ETX levels corresponding to 95% and 95% PDR are also shown in horizontal lines.

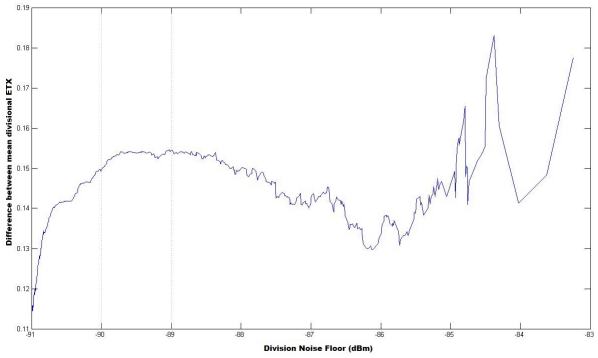


Figure 4.6: Divisions are formed at individual noise floor levels, and the corresponding differences in mean ETX between the two parts are depicted.

handling floating-point computation and storage, we only consider integer threshold values. Consequently -90 and -89 dBm are taken as potential noise thresholds.

To reinforce the findings obtained with automatic clustering, more inspection were carried out where we look directly into (NFS_{batch},ETX_{batch}) pairs. Firstly, we examine the extent of the effect ambient noise through the distribution of the (NFS_{batch},ETX_{batch}) population. To this end, the 95% percentile of ETX is plotted in red lines in Figure 4.5. The percentile line provides a reference to the

majority of ETX which is signified by value below the red line. The ETX levels corresponding to 95% and 95% PDR are also shown in horizontal lines in the figure serving as baselines above which transmission performance is considered significantly affected by ambient noise. According to Figure 4.5, the transmission success rates are overwhelmingly greater than 90% for noise floor below -89 dBm, and greater than 95% for noise floor below -90 dBm. These observations are consistent with previous findings obtained via clustering.

In an additional inspection, the ETX population is iteratively divided at individual integral noise levels. The mean ETX of either divisions are calculated and their differences are plotted in Figure 4.6. It is observed that the disparity between mean divisional ETX reaches peak level when division is made at -89 dBm, and remains at relatively high level towards -90 dBm. This indicates that the maximum distinction in average ETX performance lay between -90 to -89 dBm, reconfirming previous findings. We accordingly report the following observation on useful thresholds.

Observation 4 *The integer noise thresholds that provide the most significant separation between NFS-ETX relationship are -90 and -89 dBm, which are found to result in transmission success rates of 90% and 95%, respectively.*

4.2.6 Conclusion of observations

We believe that the transmission performance can be improved by recognising and avoiding IEEE 802.15.4 frequencies with high noise floors, according to the NFS-ETX relationship stated in Observation 1 and 2. The thresholds of ambient noise expected in Observation 3 can be potential criteria for the identification, and the useful thresholds found in Observation 4 are going to be used in different schemes later described in Chapter 6.

4.3 Noise Properties

Following observations regarding the significance of ambient noise on low-power wireless communication, the properties of noise are investigated in this section. First, the data collection configuration is specified. Then time series analysis is conducted for noise samples. The ingredients of the noise series are identified following the principle of Classical Seasonal Decomposition (CSD) technique [128], which views time series as combinations of seasonal, trend and noise factors [129]. Time-sequence plots are produced to provide a basic visual comprehension and initial intuitions of the interference patterns for different environments. Examinations on individual components are subsequently carried out to verify any preliminary discoveries. Analysis are conducted for all sixteen IEEE 802.15.4 channels and similar results have been obtained. The following discussions utilises findings on channel 12 (2410 MHz) in the interest of the clarity of demonstration.

4.3.1 Configuration of Noise Sampling

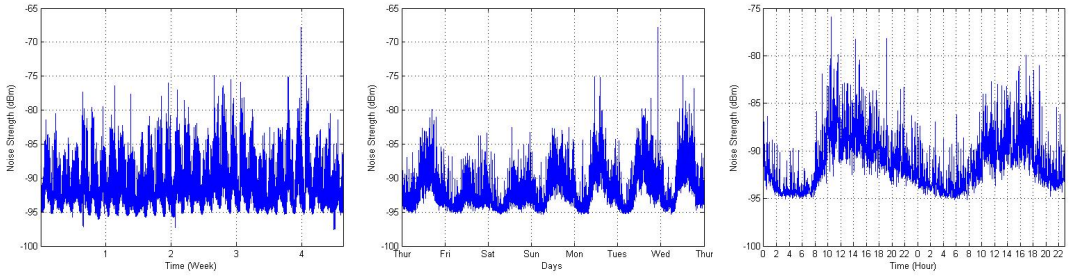
GINA motes were deployed at Location A, B and C, respectively. Instead of carrying out transmission as the experiment in Section 4.2, motes were configured to constantly sense the energy level in different IEEE 802.15.4 channels. Each sensing operation took $128\mu s$ of measurement and $12\mu s$ processing delay, totalling $140\mu s$. Groups of 128 readings were averaged into individual samples that were stored on desktops. The sampling duration for each location lasted more than a month to capture temporal variance, which are detailed in Table 4.3. Samples are attributed to the locations of their collection, and are referred to by the corresponding location IDs hereafter.

4.3.2 Overview of Noise Sample Series

The time-sequence graphs of three noise sample sets are shown in Figure 4.7, 4.8 and 4.9. Alongside the plot for the entire sampling period, segments focusing on specific weeks and days are also provided.

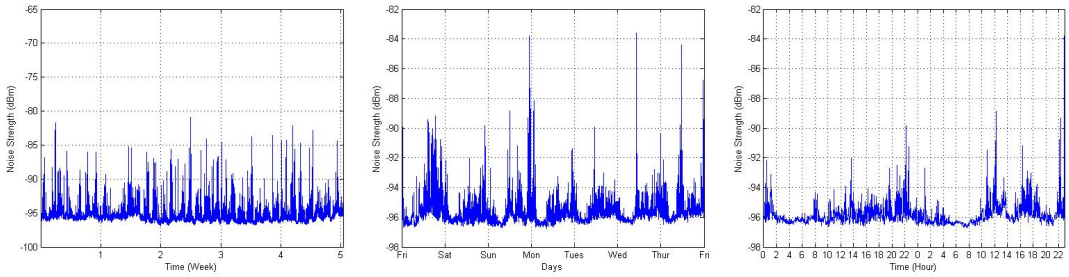
Table 4.3: Details of experiments carried out to collect noise samples. Data were respectively collected at Location A, B and C for prolonged periods of time.

Location ID	**Begin time**	**End time**	**Duration**
A	00:00 26th July 2012	00:00 6th Sep 2012	Six weeks
B	09:00 27th July 2012	15:00 31st Aug 2012	Five weeks
C	16:30 15th Aug 2012	16:30 19th Sep 2012	Six weeks



(a) Entire duration (b) Week 3 (c) Thursday and Friday

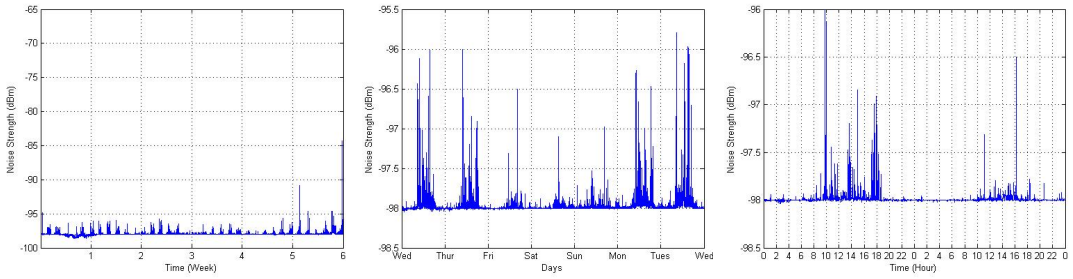
Figure 4.7: Noise levels in channel 12 at Location A at different time



(a) Entire duration (b) Week 4 (c) Monday and Tuesday

Figure 4.8: Noise readings in channel 12 at Location B at different time

Sample A exhibits repetitive cycles over 24 hours in Figure 4.7b despite occasional outliers, implying the existence of daily seasonality. Additionally, the



(a) Entire duration (b) Week 2 (c) Thursday and Friday

Figure 4.9: Noise readings in channel 12 at Location C at different time

overall noise strength is found to be weaker in weekends compared with weekdays, which is expected as few people are present in the environment in Saturdays and Sundays. Figure 4.7c focuses on Thursday and Friday of this particular week. The daily pattern is evident in the plot as noise level is predominantly above -90 dBm during typical office hours (10:00 to 18:00) and relatively low (in the region between -95 to -90 dBm) for the rest of the days.

The daily seasonality of Sample B in Figure 4.8b appears much less notable than that of Sample A. In particular, noise levels in Monday to Wednesday have notably different shapes from the other days. Such irregularity is more clearly visible in the segment of Monday and Tuesday in Figure 4.8c. Noise floor pattern between 6:00 and 8:00 on Monday is significantly more changeable than that of Tuesday; the same can also be observed between 14:00 and 16:00.

Compared with Sample A and B, Sample C exhibits the weakest seasonality. Although the sharp dips near the end of week 1 in Figure 4.9a could be attributed to recorded failures of power system in the building, noise floors in the following two weeks are still highly unpredictable. A considerable part of noise floor in week 2 shown in Figure 4.9b is largely flat; however the fluctuations in other parts appear irregular as highlighted in Figure 4.9c for Thursday and Friday of the same week.

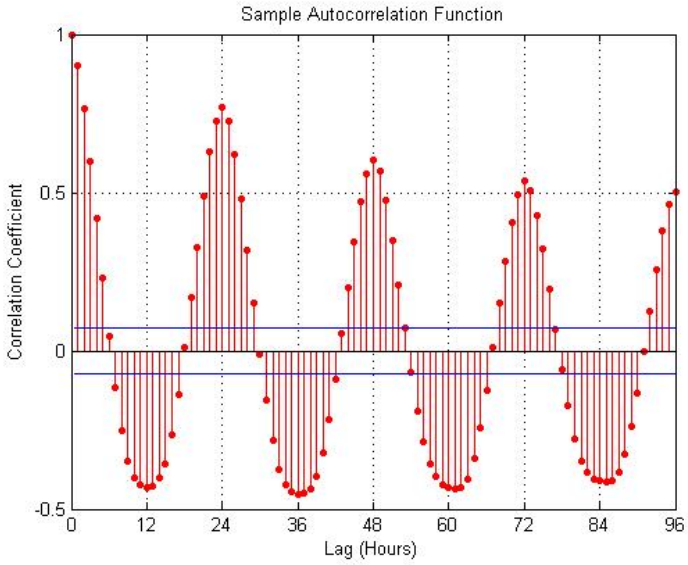
In additional to seasonality, a further inference regarding the trend component can also be drawn from the time-sequence plots. In this work trend specifically refers to disposition of continual growth or recession over the sampling duration. In spite of the fluctuations, there is a lack of palpable overall increasing or decreasing inclination in the curves of all three sequences. Thus the overall noise levels tend to be constant, *i.e.* no significant trend component is identified in the time series.

4.3.3 Seasonality

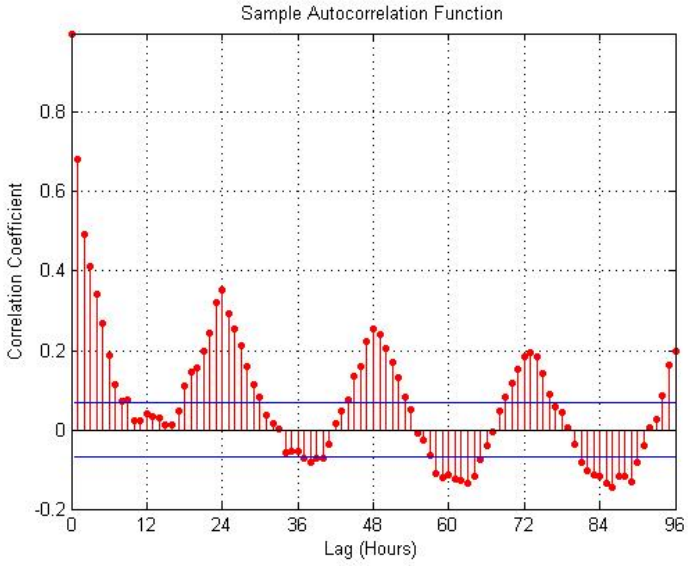
4.3.3.1 Time Domain Analysis

To investigate seasonality in the time domain, we examine the temporal autocorrelation which probes the relationship between time-lagged versions of the original series. The sample correlation function (ACF) *autocorr* of Matlab is used and the graphical presentations of its results, known as correlograms, are provided in Figure 4.10 for each sample, respectively. The extent of time-shifting is controlled by lag values shown on the horizontal axis in the unit of hour. Correlation coefficients shown on vertical axis indicate how closely the noise sequences are related to their time-lagged versions.

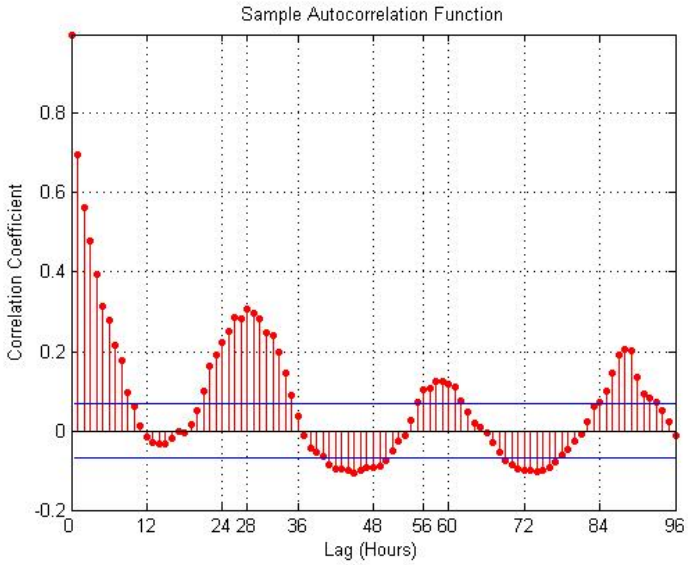
The curve in Figure 4.10a displays a resemblance to cosine function with wavelength of 24. The peaks at lags of 0^{th}, 24^{th} and 48^{th} hours expose conspicuous positive autocorrelation when the series is time-shifted by integral multiple of 24 hours. The autocorrelation is 1 at lags 0 because the series is perfectly correlated with itself. Correlations are also obvious at lags 1 and 2 because the noise tend to be consistent in short-terms hence they do not reveal the repetitive pattern we aim to discover. True seasonality is indicated at the lag of 24 hour because it provides a high correlation (approximately 0.8) after a cycle of decreasing course with a trough at the lag of 12 hours.


Sample Autocorrelation Function
Correlation Coefficient
Lag (Hours)
1
0.5
0
-0.5
0
12
24
36
48
60
72
84
96

(a)


Sample Autocorrelation Function
Correlation Coefficient
Lag (Hours)
0.8
0.6
0.4
0.2
0
-0.2
0
12
24
36
48
60
72
84
96

(b)



(c)

Figure 4.10: Autocorrelation function(ACF) plots. X axes denotes lags in unit of hours and Y axes indicates correlation coefficients. (a), (b) and (c) correspond to Sample A, B and C, respectively.

Samples B also shows peak of autocorrelation at the lag of 24 hours in Figure 4.10b but to a much weaker extent of less than 0.4. Also The symmetry of positive and negative correlation seen in Figure 4.10a is not present. In particular, the coefficient is near zero at 12 hours, which means almost no relationship exists at that specific lag. Such randomness weakens the seasonal factor and in turn partly contributes to the low correlation at 24 hours.

The correlogram of Sample C appears to also have a sinusoidal feature, however a closer look reveals that it is in fact irregular cyclic behaviour. The first significant correlation happens at the lag of 28 hours unlike the findings on Sample A and B. Moreover, peak would have occurred at the integral multiple of 28 hours such as 58 and 96 hours if there indeed existed 28-hour seasonality. However it is apparently not the case in Figure 4.9c where subsequent peaks occur at lags of 60 and 88 hours. As a consequence, no seasonality of noise level is

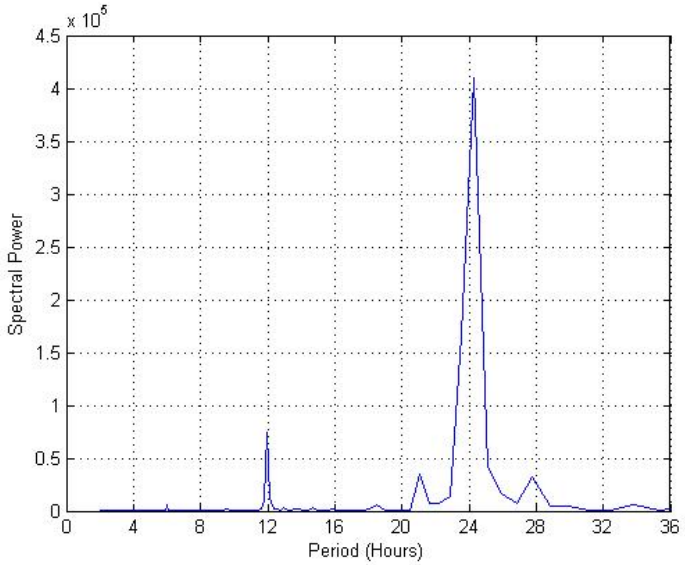
identified for Sample C.

4.3.3.2 Frequency Domain Analysis

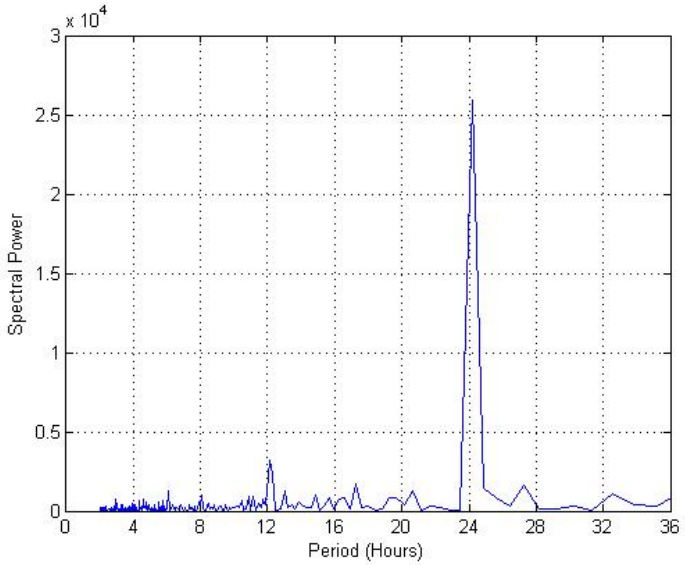
Seasonality is also examined through spectral density analysis in the frequency domain. Signals are interpreted as combinations of a series of sine waves of different amplitude, periods and phases. By examining the Power Spectral Density (PSD) of the samples, the significance of individual sinusoidal ingredients can be revealed which represents the power of seasonal factors with corresponding periods in the original sample. The resulting PSD plots in Figure 4.11, also known as periodograms, of the observed noise level are produced in Matlab.

The periodograms for three sample sets are displayed in Figure 4.11. The horizontal and vertical axis denote the periods and significance of individual spectral components, respectively. An overwhelmingly high power at 24 hours is observed in Figure 4.11a which discloses the fact that the predominant ingredient in Sample A has the period of 24 hours. This confirms our earlier findings that the noise floor has a daily seasonality. The periodogram for Sample B displays the similar attribute in Figure 4.11b but the absolute power of the peak at 24 hour is lower by the factor of 10 compared with Sample A, which indicates a weaker periodicity. In a notable comparison, the prominence of the 24-hours period is not present in Figure 4.10c for Sample C. The highest power no longer appears at 24 hours but rather 30 hours. However this is not sufficient to prove 30-hours seasonality mainly because of two reasons. First, there exist multiple considerable peaks between 24 and 36 hours which indicate the pattern tends to be irregular cyclic behaviour rather than a seasonal repetition. Secondly, the absolute power at 30 hours is substantially lower than power of peaks found in Figure 4.11a and 4.11b.

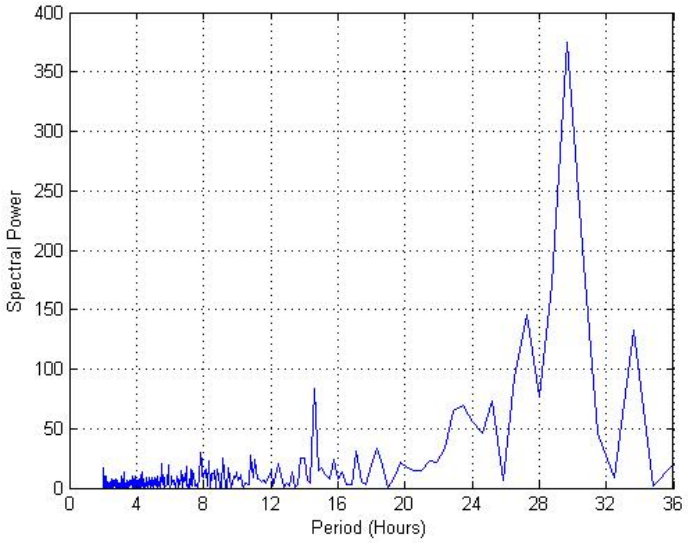
The variation in the significance of seasonal factor among three noise sequences is more clearly demonstrated Figure 4.12 where the periodograms of three samples are depicted together in the same scale. The blue dot-dashed line corresponds to Sample A and green dashed line with triangle is for Sample B; Sample C is


x 10^5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Spectral Power
0
4
8
12
16
20
24
28
32
36
Period (Hours)

(a)


x 10^4
3
2.5
2
1.5
1
0.5
0
Spectral Power
0
4
8
12
16
20
24
28
32
36
Period (Hours)

(b)



(c)

Figure 4.11: Periodogram of channel 12 at all locations. (a), (b) and (c) correspond to Sample A, B and C, respectively.

represented with red line with cross marks. It is clear in the plot that the spectral power peak is much more powerful in Sample A than in Sample B. And the fact that Sample C is almost flat suggests that there is a significantly powerless, if any, seasonal component. Accordingly only Location A out of three experimental environments shows apparent seasonal factor. This finding also indicates that seasonality of ambient noise is not a universal but environment-specific feature.

4.3.4 Trend

To have a better view of potential trend factor, the seasonal factor needs to be removed. Classical decomposition uses weighted moving average technique to cancel the effect of seasonality [130] and the results are seasonally-adjusted version of the original series. Figure 4.13a and 4.13b illustrate the seasonally-adjusted series of noise Samples A and B. Sample C is exempt from the process since it does not have clearly identified seasonal factor.

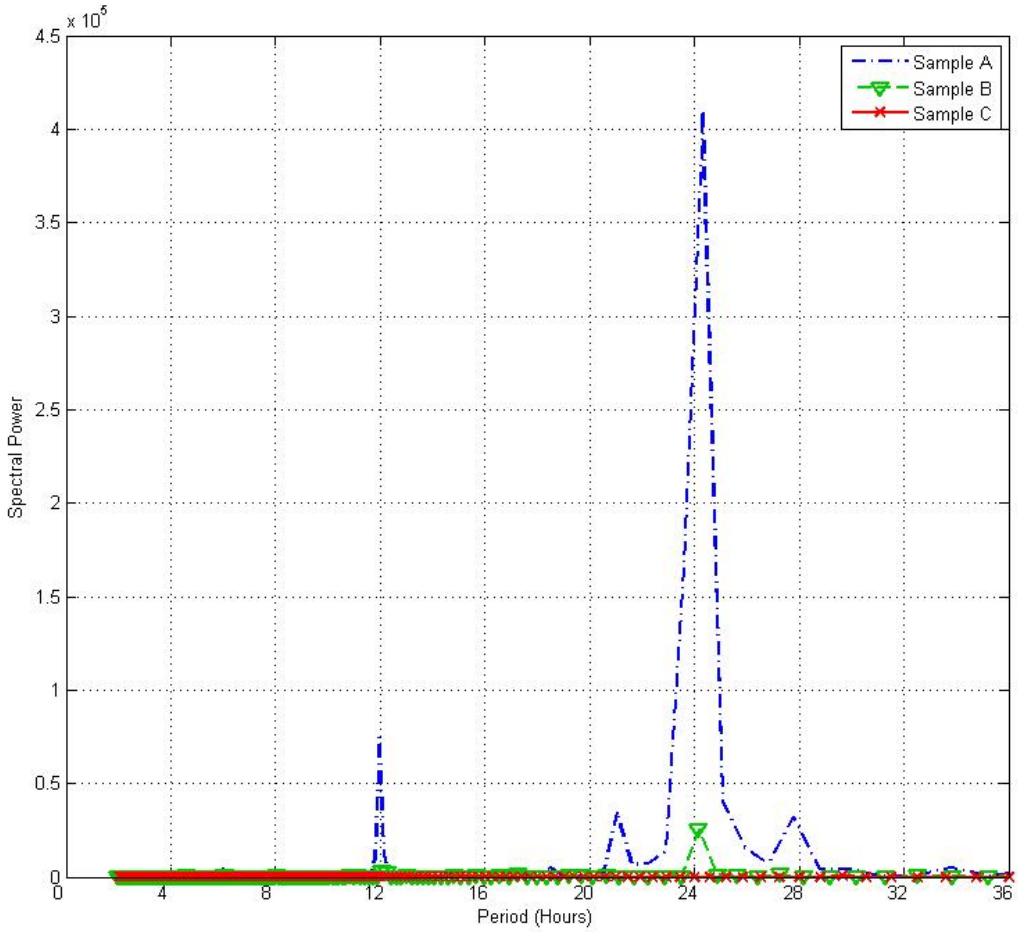
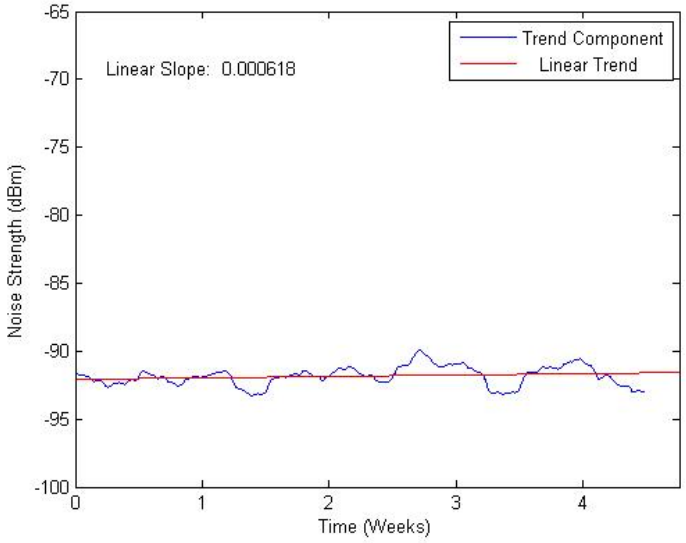
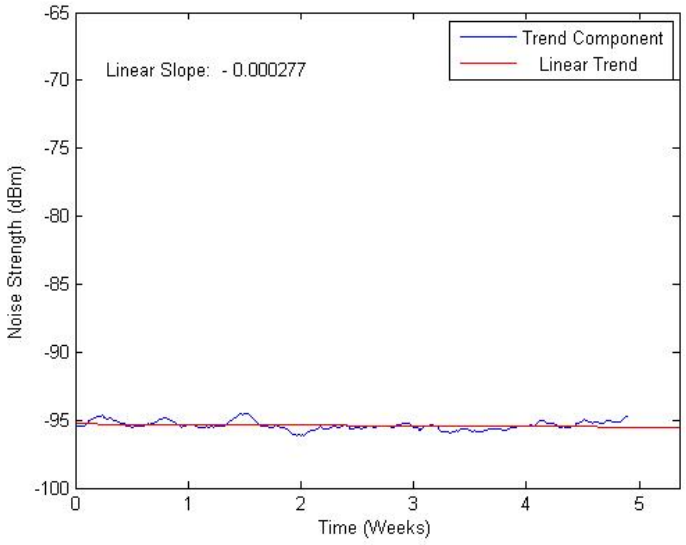


Figure 4.12: Periodograms for 3 sample sets using the same scale. Sample A, B and C are represented with blue dot-dashed line, green dashed line with triangle and red solid line with cross, respectively.

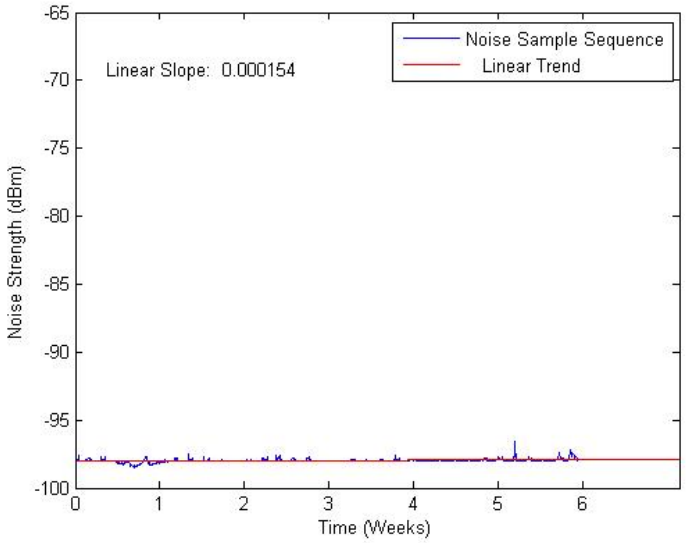
Linear regression is applied to the series to help identify trending effect. The resulting fits are almost flat trendlines throughout the duration, confirming previous inference that the overall increasing or decreasing trend is absent. Additional verification can be drawn from periodograms in Figure 4.11. In case of the presence of long term trend, there should be a significantly powerful peak associated with certain period much longer than 24 hours; nevertheless, no such phenomenon


Trend Component
Linear Trend
Linear Slope: 0.000618
Noise Strength (dBm)
Time (Weeks)
-65
-70
-75
-80
-85
-90
-95
-100
0
1
2
3
4

(a)


Trend Component
Linear Trend
Linear Slope: - 0.000277
Noise Strength (dBm)
Time (Weeks)
-65
-70
-75
-80
-85
-90
-95
-100
0
1
2
3
4
5

(b)



(c)

Figure 4.13: Trend component of three noise sample sets. (a), (b) and (c) correspond to Sample A, B and C, respectively.

has been observed in Figure 4.11.

4.3.5 Remarks on Noise Attributes

Analysis in this section revealed some basic properties of ambient noise which were found to be environment-specific. Relatively notable seasonal factor was found in Sample A. This can be attributed to different natures of noise sources. At Location A, the dominant noise sources were the proximately installed wireless routers around the workplace, as well as clearly visible WLAN signals from adjacent office buildings. As these WLANs were primarily used by people working in these premises, the noise emission level showed a rough pattern of office hours despite some inevitable irregularity due to unpredictable nature of human activities. Location B, on the other hand, was a residential flat where noise mainly came from the wireless Internet access point shared by occupants. Since the residents included office workers, housewives, students and job seekers, the utilisation profile of WLAN was much less predictable compared with Location A,

resulting in a weaker seasonal feature. In comparison, Sample C exhibited no obvious seasonality. Several reasons could have contributed to this finding. Firstly, most departmental computers connected to Internet via wall sockets which significantly reduced the wireless interference. And the lack of a predominant noise source gave rise to more randomness caused by ad hoc wireless communication between mobile devices. Moreover, the sampling period coincided with vacation time and the Olympics, as a result usual daily schedule might have been affected. In addition, in none of the samples was any obvious trend identified, and indeed there had not been any factors, to our best knowledge, that would have caused long-term increasing or decreasing tendency such as regular addition or removal of wireless devices.

4.4 Summary

This chapter in the first part provided supporting evidence for the decision of using ambient noise as indicators of channel quality for A-TSCH. We investigated negative effect of ambient noise on transmission success rates and observed that performance varies among IEEE 802.15.4 channels because of their different noise floors. Then the existence of noise threshold of was anticipated and through statistical analysis two integral candidates were identified. Recognising the significance of ambient noise, the second part of this chapter subsequently examined the properties of noise samples using time series analysis approaches. Seasonality of different significance was found in Sample A and B, whilst no trend factor was observed. Findings in this chapter will assist later chapters, especially Chapter 5 and 6.

Chapter 5

Channel Quality Estimation

This chapter discusses the Channel Quality Estimation (CQE) module of A-TSCH which is responsible for forming suitable knowledge about the desirability of specific channels for wireless communication. In Section 5.1, different techniques are considered for a simple lightweight quality estimator module for A-TSCH. Subsequently the choice of estimator parameters is discussed in Section 5.2. Then the performances of estimators are investigated through simulation in Section 5.3. Finally this chapter is summarised in Section 5.4.

5.1 Noise Estimators

Motivated by findings in Chapter 4, A-TSCH uses ambient noise floor as the main indicator of channel quality. However, some practical limitations should be considered. On the one hand, noise readings are subject to sudden momentary fluctuation in noise energy strength and may not accurately reflect channel quality over longer periods. On the other hand, noise samples in A-TSCH are periodically collected in dedicated NF timeslots. These temporal gaps between readings can further affect their accuracy. To mitigate this problem, the CQE module employs estimation techniques and the Exponential Smoothing (ES), Kalman filter (KF) and Exponential Smoothing aided by Kalman filter (KFES) are discussed in this section.

5.1.1 Exponential Smoothing

Noise floor measurements are time series from a statistical perspective since they are data points at uniform intervals [130]. Exponential Smoothing (ES) is a popular estimation technique for time series, known for its effectiveness and relative simplicity [131, 132]. There are different variants of ES for different characteristics of time series. In this work, we consider the *Single* ES which is the most basic form, and the *Seasonal* ES due to the seasonal factor has been observed in Section 4.3.

5.1.1.1 Single Exponential Smoothing

Single ES, also known as Simple ES, is the most basic form of ES which does not assume the presence of trend or seasonal ingredients of time series. This method is characterised by its low computational overhead which makes it suitable for low-power wireless devices.

$$\hat{Y}_{t+1|t} = \alpha O_t + (1-\alpha) Y_{t|t-1} \tag{5.1}$$

The operation of Single ES is generalised in Equation (5.1). O_t denotes the observation at time t and $Y_{t|t-1}$ is the estimate for time t previously generated at time *t-1*. The estimate for time *t+1* at t, denoted by $\hat{Y}_{t+1|t}$, is the weighted average of O_t and $Y_{t|t-1}$, subject to smoothing coefficient $\alpha \in (0, 1)$. Estimates in Single ES are calculated entirely on the most recent estimate and the current observation, ensuring both modest computation and storage overhead for low-power devices.

5.1.1.2 Seasonal Exponential Smoothing

Because the noise series exhibit certain level of seasonality in Section 4.3, we also consider Seasonal ES which takes into account the seasonal factor. Estimation procedures using Seasonal ES is provided in Equation set (5.2), which represents

a special case of Holt-Winters method [133] with the trend term removed according to the taxonomy in [134].

$$\hat{Y}_{t+1|t} = L_t + S_{t-p+1} \tag{5.2a}$$

$$L_t = \alpha(O_t - S_{t-p}) + (1-\alpha)L_{t-1} \tag{5.2b}$$

$$S_t = \gamma(O_t - L_t) + (1-\gamma)S_{t-p} \tag{5.2c}$$

As shown in Equation (5.2a), the estimate for time *t+1* yielded at *t*, $\hat{Y}_{t+1|t}$, is the combination of non-seasonal factor L_t and seasonal component S_{t-p+1}, where p is the length of season. The non-seasonal factor L_t and seasonal component S_{t-p} are computed based on Equation (5.2b) and (5.2c), respectively. L_t, is the weighted average of seasonally-adjusted observation $(O_t - S_{t-p})$ and the previous estimate L_{t-1} at time *t-1*, depending on coefficient α. The seasonal factor is computed as the weighted averaged of seasonal components at the current time $(O_t - L_t)$, and an entire season ago (S_{t-p}), depending on an additional seasonal coefficient γ.

5.1.2 Kalman Filter

Kalman filter (KF) [135] is an alternative method considered for the CQE module. Compared with ES estimators which use static coefficients, the weighing factors in KF are continuously adjusted [136] to achieve gradual convergence to the true system state.

The KF process is illustrated in Figure 5.1. Functions in the figure are simplified based on the standard form defined in [137] to suit our specific case: firstly, the monitored noise floor is one-dimensional information therefore reducing matrix factors to scalar coefficients; secondly, since no definitive model is available for channel noise floor, the overall noise level is assumed to be largely constant

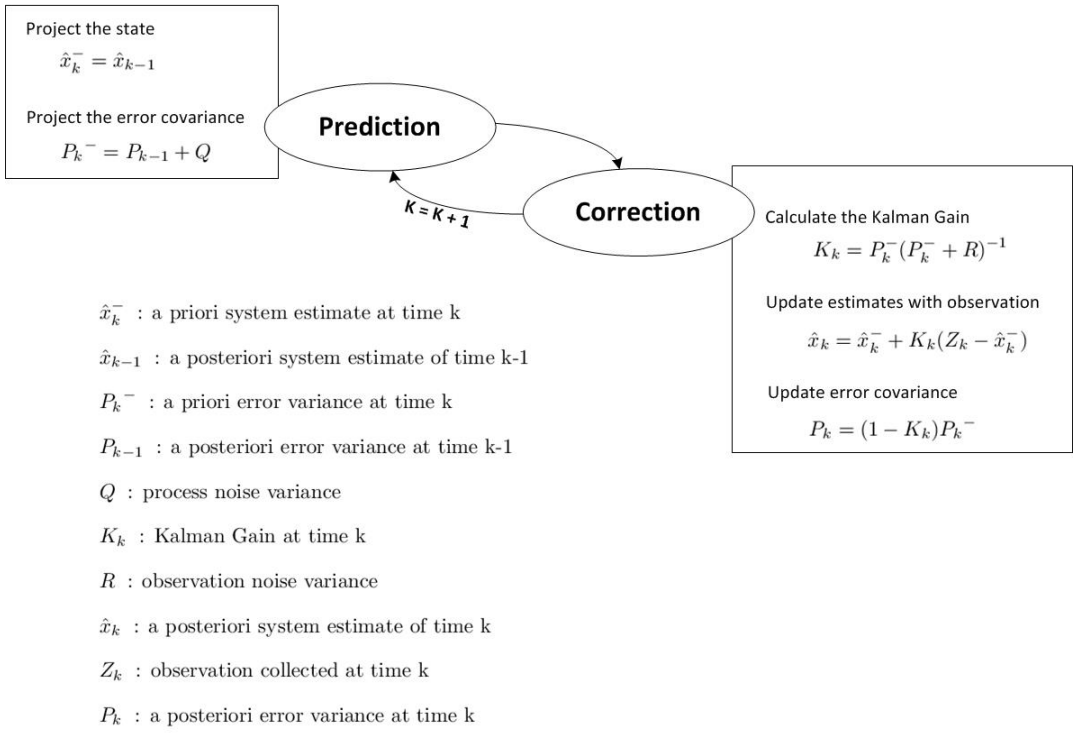


Figure 5.1: Operation of Kalman filter

and hence the system model factor is omitted.

The process of KF is often interpreted as an iterative cycle of Prediction and Correction phases, as illustrated in Figure 5.1. Starting with the Prediction phase, KF firstly generates a projection of the system state which in our case is the channel noise floor. After a new observation of noise floor is obtained, the control is passed to the Correction phase. Then the projected state is updated according to observation to form an estimate. The updated state is used in the Prediction phase of the next cycle as KF cycles proceed recursively.

5.1.3 Exponential Smoothing aided by Kalman Filter (KFES)

An additional type of estimator, Exponential Smoothing aided by Kalman Filter (KFES), is devised in this work aiming to combine the benefit of ES and KF. As an extension to Single ES, coefficient α of KFES is dynamically adjusted using KF based on the assumption that the most suitable smoothing coefficient tends to be stable in a long term.

Algorithm 5.1 The algorithm of KFES estimator

Data :
N_k : Noise observation at time k
α_{k-1} : Coefficient used at k-1 (to generate EST_k)
EST_{k-1} : Estimated noise level for time k-1
Result :
EST_{k+1} : Noise level estimate for time k+1

1: **function** MAIN
2: **while** (slotType == NF) AND (N_k available) **do**
3: Retrieve N_k
4: $bestAlpha$ = BESTCOEFFICIENT(N_k, EST_{k-1})
▷ Retrospectively get the best α for last step
5: α_k = KF(bestAlpha) ▷ Get estimate of optimal α
6: EST_{k+1} = ES(N_k, α_k) ▷ Get forecast of noise level
7: **end while**
8: **end function**

9: **function** BESTCOEFFICIENT($noise$, est)
10: Find $\check{\beta}$ that minimizes $|N_k - \textbf{BESTCOEFFICIENT}(EST_{k-1}, \beta)|$
11: Return $\check{\beta}$
12: **end function**

This KFES estimator is described in Algorithm 5.1. Note that details of function *ES* and *KF* are omitted from the algorithm listing since their operations have been discussed previously. For KFES, it is assumed that noise samples truthfully reflect the momentary states of noise floor. Thus it is possible to determine the α

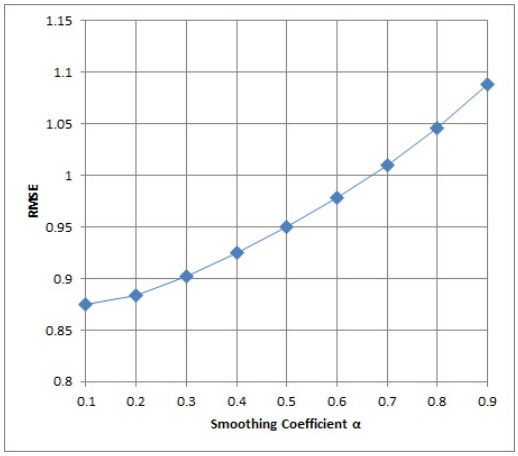
which could have led to optimal estimates in the previous step. This retrospective information is denoted by *bestAlpha* and used as the correction value for KF. The estimate generated by KF is taken as the ideal α and subsequently used to yield noise estimates.

5.2 Tuning Channel Quality Estimators

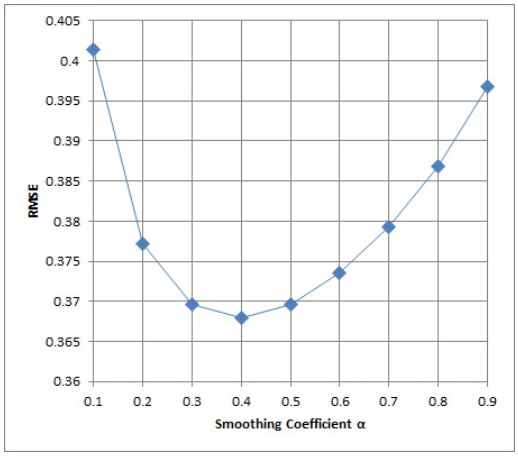
Both ES and kF are controlled by a number of parameters that are typically associated with specific systems that the estimators are applied to. In this section we examine the tuning of parameter values for CQE estimators. Investigations are carried out through simulation. Specifically, estimators are implemented in Matlab and Sample A, B and C introduced in Section 4.3 are used. To facilitate the estimator tuning and the performance investigation afterwards, the *out-of-sample* test approach is employed. Out-of-sample tests divide certain sample into two halves [138]: the first portion is used for parameter selection in this section and the second is utilised to test estimator performance in Section 5.3. Estimators are tuned based on the accuracy of estimation in terms of Root Mean Square Error (RMSE) between the original noise floors and resulting estimates. Then the set of parameters that delivers the most accurate estimates is identified and selected for further comparison and deployment.

5.2.1 Exponential Smoothing Coefficients

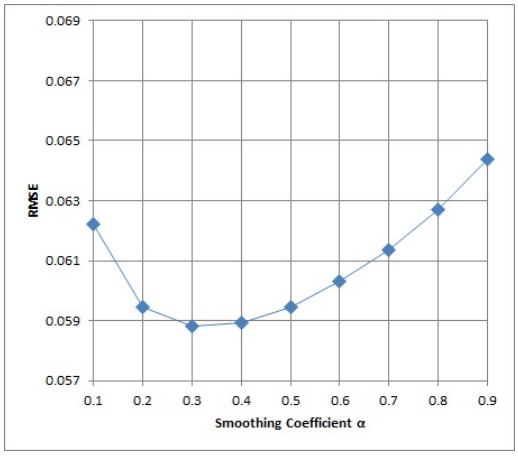
For tuning the basic smoothing coefficient α and the seasonal coefficient γ, nine candidate values are evenly extracted between *0.1* to *0.9* at a step of *0.1*. The estimation accuracy of Simple ES is measured and illustrated in Figure 5.2. The basic coefficient α yielding best estimates are found to 0.1, 0.4 and 0.3 for Sample A, B and C, respectively. Adopting these corresponding α values, the accuracy of Seasonal ES estimator using different γ is also tested. As the results in Figure 5.3 shows, the best-performing seasonal smoothing coefficients are 0.2, 0.3, and 0.1 for Sample A, B and C, respectively, due to specific characteristics of the



(a)

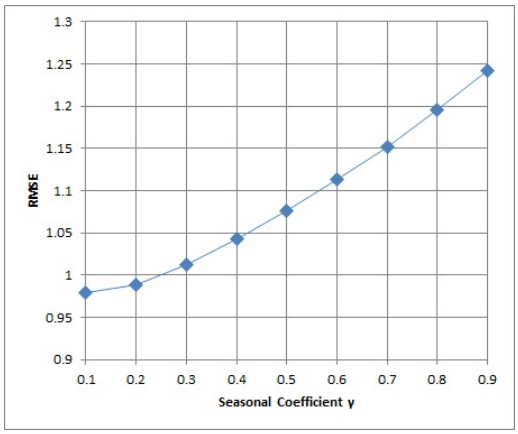


(b)

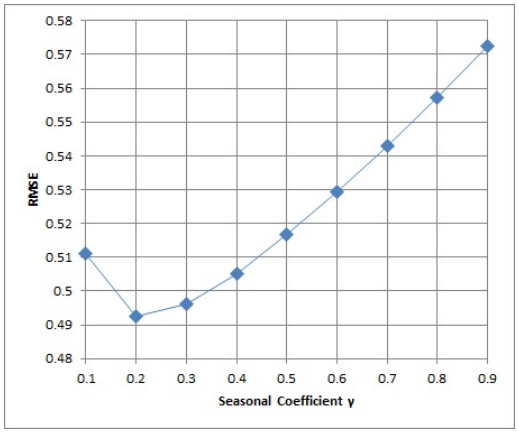


(c)

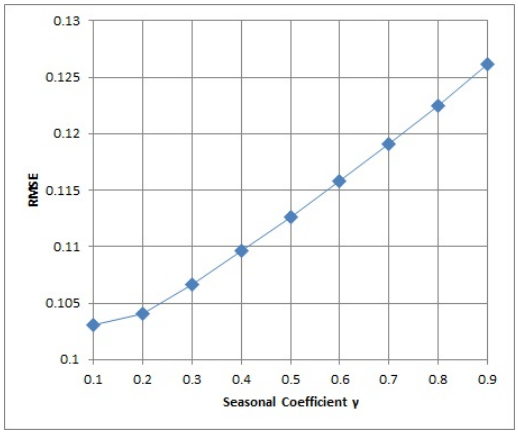
Figure 5.2: Estimation accuracy of different basic smoothing coefficients α for Samples (a) A, (b) B and (c) C.



(a)



(b)



(c)

Figure 5.3: Estimation accuracy of different seasonal smoothing coefficients γ for Samples (a) A, (b) B and (c) C.

environments present at the time of sample collection.

5.2.2 Kalman Filter Error Variance

Kalman filter requires that the observation error variance R and process measurement error variance Q are determined in advance. Observation error variance R is typically related to intrinsic characteristics of equipments. In this work we assume R corresponds to the variance of RSSI readings captured by the mote. According to the specification of the Atmel tranceiver used in GINA motes, the maximum difference between detected and true energy strength is ± 6 dB [139]. Based on the Empirical Rule [140] for normal distribution, the boundaries of $\mu - 6\sigma$ and $\mu + 6\sigma$, that is, 6 standard deviations σ from the mean μ, allows for an outlier probability of less than $2 * 10^{-9}$. Therefore it is sufficient to assume 6σ is equivalent to the measurement error of 6 dB. Another reason for this choice is that $R = 1$ can be deduced from the fact that $Var = \sigma^2$ by letting $6\sigma = 6$, which greatly reduces the computational complexity for low-power wireless motes which are not efficient in floating-point calculations [141].

Process error variance, on the other hand, denotes our confidence in the understanding of the system model. Q in most practical cases is unknown [142] and typically cannot be directly observed [137]. This is particularly pertinent to the case in this work because a general assumption of constant system has been made, in which the confidence is fairly uncertain. The method previously used to determine exponential smoothing coefficient is again adopted. Unlike the α, Q does not have a fixed range. Consequently the function minimisation of Matlab is employed to discover Q value that yields the smallest root-mean-square error (RMSE) in 16 channels respectively. The results are summarized in Table 5.1.

Table 5.1: Q values for 16 IEEE 802.15.4 channels obtained through simulations on three samples

Channel / **Sample**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
A	0.008	0.008	0.010	0.011	0.017	0.017	0.013	0.017	0.020	0.017	0.013	0.016	0.013	0.013	0.009	0.008
B	0.220	0.200	0.220	0.245	0.147	0.510	0.368	0.330	0.659	0.578	0.100	0.129	0.069	0.114	0.198	0.164
C	0.253	0.120	0.102	0.021	0.011	0.109	0.181	0.119	0.210	0.015	0.365	0.282	0.290	0.145	0.015	0.036

5.3 Estimator Performance

In the continuing stage of out-of-sample test, the performance of the estimators are examined through simulation using parameters determined above. The second portion of noise samples are used for simulation and estimator accuracy is measured in terms of Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) between the captured and estimated noise floors.

Table 5.2: Estimator accuracy observed in simulations. The accuracy for three sample sets is assessed in terms of Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE), respectively. The results achieved without using any estimator are also provided as reference.

Error Type	**Estimator / Sample**	**None**	**ES**		**KF**	**KFES**
			Single	**Seasonal**		
RMSE	**A**	1.002	0.788	0.943	0.786	0.788
	B	0.334	0.304	0.380	0.305	0.305
	C	0.075	0.067	0.084	0.067	0.070
MAPE	**A**	1.1%	0.86%	1.03%	0.86%	0.86%
	B	0.36%	0.33%	0.4%	0.33%	0.33%
	C	0.08%	0.07%	0.09%	0.07%	0.08%

Results presented in Table 5.2 are average errors over all 16 channels. Data in first column of the table are calculated without using any estimators, which provide a baseline to highlight the improving effect of using estimators than directly using the raw noise readings.

For two ES variants, the Simple ES outperforms Seasonal ES with a notable margin. This result is expected for Sample B and C because of their relatively

weak seasonality. However, the inferior outcome of Seasonal ES for sample A is possibly the result of the fact that the initial seasonal factor is derived from the suggestion in [130] which as the author states is a simplistic form and may not perform well. Another possible reason is that seasonality observed in Sample A is an overall regularity that is only significant over longer durations. Therefore the seasonal factor might be useful for estimating, for example, hourly average noise floors, but may not be effective in improving accuracy of more frequent estimation. The Kalman filter, despite largely simplified according to our specific case, provides similar accuracy as the Simple ES. Comparable results are obtained with KFES as well.

Overall, all estimators provide better accuracy in determining the future noise floor than using unprocessed raw noise readings. However, seasonal ES produces the least improvement whilst incurring increased computational overhead. Therefore it will not be considered further for our implementation of A-TSCH. The performance of the remaining three are comparable.

5.4 Summary

In this chapter, we discussed the Channel Quality Estimation (CQE) module which transforms ambient noise readings containing momentary fluctuations into channel desirability knowledge more suitable for A-TSCH. Different estimation techniques were considered, and out-of-sample tests were carried out for parameter tuning as well as performance comparison. In particular, Simple ES, KF and KFES were found to provide comparable estimation accuracy and will be further investigated through hardware implementation in subsequent chapters.

Chapter 6

Cooperative Blacklisting

In this chapter, we discuss in detail the Cooperative Blacklisting (CB) module which is at the core of A-TSCH framework. Section 6.1 statistically demonstrates that blacklisting can notably mitigate the adverse effects of noise and improve successful delivery rates. Different cooperation strategies for blacklisting are reviewed before the approach adopted by CB is described in Section 6.2. Subsequently, Section 6.3 details the process for generating blacklists, followed by the algorithm of applying blacklists to channel hopping in Section 6.4. Finally a summary of this module is provided in Section 6.5.

6.1 Potential Benefit of Blacklisting

Motivated by observations in Section 4.2, A-TSCH employs blacklisting in the anticipation of improved transmission success rates. In this section, we conduct a statistical investigation into the potential benefit of blacklisting by simulating its operation using existing data traces with different configurations.

6.1.1 Test Settings

The NFS_{batch} and ETX_{batch} values of communication traces introduced in Section 4.2 are used for this section. Because packets were transmitted in batches

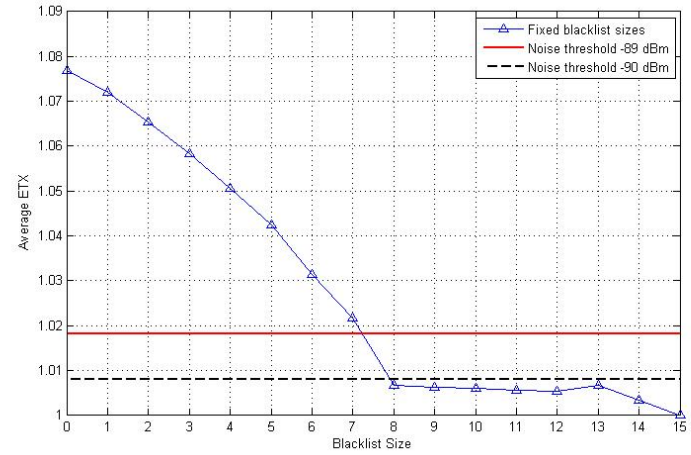
using different frequencies in a round-robin fashion, every group of 16 consecutive (NFS_{batch},ETX_{batch}) pairs corresponds to a round of all 16 IEEE 802.15.4 channels. Accordingly the records can be transformed into a series of such groups which form the basic units for our test steps.

Blacklist used in step $i+1$ is derived from NFS of the i^{th} step. The size of blacklist is denoted by K, meaning the ETX_{batch} of these K channels are eliminated from the calculation for this specific step. Consequently, the resulting ETX of each test step is calculated on the ETX_{batch} of the remaining $16-K$ channels in each groups of 16 (NFS_{batch},ETX_{batch}) records.

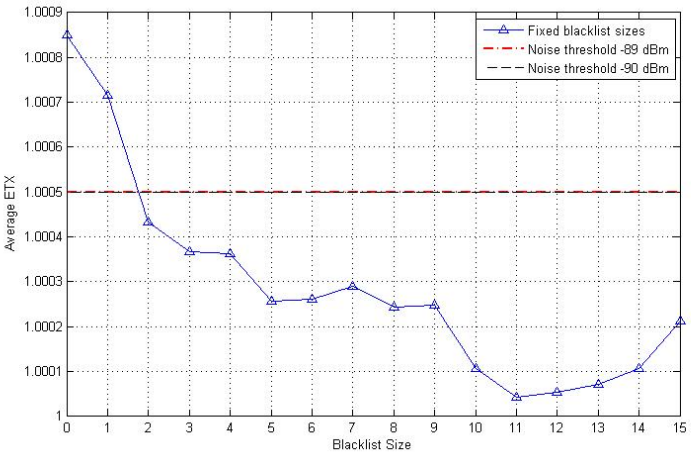
Recalling Observation 3 in Section 4.2 which states ambient noise level only significantly affect ETX if greater than certain threshold, an alternative blacklisting method is to dynamically identify and exclude channels with noise floors exceeding the thresholds, instead of imposing fixed blacklist sizes regardless of their conditions. Observation 4 suggests that -89 or -90 dBm are potential candidates for the integral noise threshold. Accordingly tests are also conducted using blacklisting based on both thresholds, respectively. In any given test step, channels whose NFS_{batch} are greater than the threshold are excluded in the next step.

6.1.2 Test Results

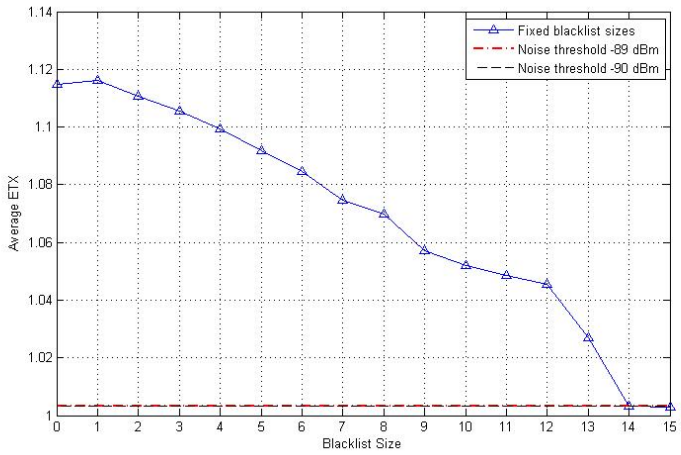
Test results obtained using three data traces are illustrated in Figure 6.1a, 6.1b and 6.1c, respectively. Triangles on blue solid lines correspond to results obtained with different blacklist sizes $K \in [0, 15]$ where top K channels in the noise floors ranking get blacklisted. In particular, blacklist size 0 denotes unaltered standard channel hopping without blacklisting, whilst size 15 blocks access to all but one channel and hence is equivalent to single-channel communication.


Fixed blacklist sizes
Noise threshold -89 dBm
Noise threshold -90 dBm
Average ETX
Blacklist Size
1.09
1.08
1.07
1.06
1.05
1.04
1.03
1.02
1.01
1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

(a)


Fixed blacklist sizes
Noise threshold -89 dBm
Noise threshold -90 dBm
Average ETX
Blacklist Size
1.0009
1.0008
1.0007
1.0006
1.0005
1.0004
1.0003
1.0002
1.0001
1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

(b)



(c)

Figure 6.1: The potential effect of blacklisting. Horizontal axis denotes number of excluded channels; vertical axis is average ETX. Triangles on the blue solid line represent ETX achieved with corresponding blacklist sizes. Results of blacklisting using noise thresholds of -89 and -90 dBm are shown in solid red and dashed black horizontal lines, respectively. (a), (b) and (c) correspond to Sample A, B and C.

It can be observed that ETX achieved with blacklist sizes greater than zero are uniformly lower than that of standard channel hopping. Specifically, significant reduction of ETX is obtained as the blacklist size grows from 0 to 8 as shown in Figure 6.1a. This can be attributed to the fact that there existed eight overwhelmingly noisy channels at Location A, according to Figure 4.3a in Section 4.2. Therefore transmission was unreliable in these channels and the overall ETX drops notably as they are excluded from the hopping sequence. Similar observations can be made for traces of Location B and C as well. It is also found that ETX does not monotonically decrease as the blacklist size increases. This is because noise information in i^{th} step does not necessarily remain accurate in step $i + 1$, underlining the dynamic nature of ambient noise.

There are also variations among test results for three experimental locations. In Figure 6.1a, blacklist size 15 provides the best ETX which indicates that there exists a channel at Location A which constantly provided the best transmission

condition. Consequently using this channel exclusively would in theory yields the optimal performance. In practice however, it is not suitable to use blacklist size 15: first, it effectively disables channel hopping and the system become vulnerable to single point of failure if the channel suddenly deteriorates. Furthermore, the existence of an overwhelmingly desirable channel is specific to the particular environment at the particular time of experiment, which is expected to represent a special case rather than normal scenarios.

Table 6.1: Performance of blacklisting using noise thresholds. Results are measured in ETX and the reduction in ETX compared with non-blacklisting channel hopping.

Location	**A**		**B**		**C**	
Threshold (dBm)	**-89**	**-90**	**-89**	**-90**	**-89**	**-90**
ETX	1.0182	1.008	1.0005	1.0005	1.0032	1.0032
Reduction	6%	7%	0.04%	0.04%	11.61%	11.61%

Results of blacklisting using noise thresholds are provided in Table 6.1, and also illustrated using the solid red and dashed black horizontal lines in Figure 6.1. At Location A, thresholds -89 dBm and -90 dBm deliver an reduction in ETX by 6% and 7%, respectively. As reflected in Figure 6.1a, the performance of the -89 dBm threshold is better than half of the fixed blacklist sizes whilst the -90 dBm threshold reduces ETX to an extent comparable with best that fixed blacklist sizes can achieve.

Reduction of ETX at Location B is not as significant. This is due to the fact that the majority of channels in that environment have noise floors below either of the thresholds as revealed in Figure 4.3c in Section 4.2, hence limited room for blacklisting to make significant difference. This does not affect the effectiveness of using noise threshold for blacklisting since ETX in this case are generally low

in all channels and the difference in the order of 0.0001 only has minimal effect on the overall transmission reliability. In comparison, most notable enhancement in ETX is observed in results for Location C because many channels have noise floors exceeding the thresholds during the experiments.

The above results provide supporting evidence that blacklisting can improve transmission performance of standard channel hopping. Furthermore, it is found that blacklisting using noise thresholds can deliver performance better than or at least comparable to that achieved with fixed blacklist sizes. Since the ambient noise is essentially environment-specific and time-varying, we believe it is a more appropriate approach for blacklisting. Noise thresholds identified in Observation 4 are expected to be suitable criteria for blacklisting. In Chapter 7, the benefit of blacklisting will be tested through implementation.

6.2 Blacklisting Strategies

In this section, we overview some strategies for blacklisting and describe the method used for CB module in A-TSCH. A technique analogous to blacklisting is called *whitelisting* [143, 45] which produces a whitelist containing admissible channels. Although blacklisting and whitelisting are essentially the same in a sense that they both separate one group of channels from the rest, they imply different strategies for communication adaptation. Whitelisting only allows using channels that are recognised as admissible; in comparison, all channels are by default admissible with blacklisting until they are explicitly disabled. In other words, blacklisting aims to make adaptation in a more conservative manner that gives priority to the stability and robustness of channel hopping.

Blacklists generated by individual nodes are typically subject to spectral peculiarity of their specific locations, and this spatial diversity should be taken into account. A-TSCH addresses this problem by employing cooperative blacklisting which enables blacklists to be decided based on information from more than one

node. Similar to the cooperative spectrum sensing discussed in Chapter 2, there are different cooperation strategies. In this section, we consider the *global*, *link-wise*, and *cluster-wise* approaches.

The global strategy employs the philosophy of centralised spectrum sensing cooperation model discussed in Section 2.3.1.1. With this approach, a universal blacklist for the entire network is produced at the gateway node which receives local channel quality reports from other nodes. The most notable advantages of this strategy are the concentration of computational complexity and the reduced synchronisation overhead for using a uniform blacklist. Because the gateway node is typically powered by stable source, the overhead incurred by the computation of blacklists tend not to pose energy drainage problem. The global cooperation also minimises the potential risk of misaligned selective channel hopping sequence caused by desynchronised blacklists.

However, the global strategy is subject to several potential problems. First, global cooperation in a network scale may incur excessive communication overhead. Local reports from nodes at the edge of the network need to be forwarded many times before reaching the gateway, which intensifies the power consumption of nodes en route and leads to delays in gathering information for cooperation. Second, the relative simplicity of global cooperation tends to result in a lack of accuracy and granularity to accommodate communication at specific locations. The purpose of cooperation is to produce blacklists that capture channel desirability of more than a single node; however the spatial diversity of wireless medium condition makes it unrealistic for a single global blacklist to be sufficient, particularly when the network covers an extensive area.

Alternatively, the link-wise strategy can address the accuracy issue of the global cooperation. In this model, cooperation is carried out on the basis of communication links. Both sender and receiver exchange their independently proposed blacklist candidates and jointly decide upon the blacklist to use on

the link. Compared with global cooperation, the link-wise strategy ensures the blacklists in place reflect the channel desirability for communication at the specific location. This improvement, however, comes at the expense of increased overhead for storage and computation. Nodes need to maintain as many blacklists as the number of neighbours they communicate with, and cooperative operations are required for each link separately.

A trade-off between the global and link-wise strategies is provided by the hierarchical cooperative model which groups nodes into clusters as discussed in Section 2.3.1.2. One node in every cluster is appointed the clusterhead which acts as the controlling entity over the its cluster members. Instead of sending blacklist candidates to the gateway or exchange them with communicating peers, nodes report to the head of their clusters which formulate the blacklist for the entire cluster. Hence, in this cluster-wise cooperation, each cluster is effectively a subnet using global cooperation within its own boundary.

The CB module of A-TSCH adopts this cluster-wise strategy since it exhibits a range of favourable characteristics compared with the global and link-wise approaches. First, exchange of candidate reports confined within individual clusters is more manageable in terms of the transmission delay and power consumption. Second, cluster blacklists provide better accuracy in representing local spectral condition than a global blacklist. Although cluster blacklists are potentially less accurate as link blacklists, the chances of apparent intra-cluster spectral variation can be minimised if proper clustering criteria are used. The potential issue can be further mitigated with suitable decision-making algorithms. Third, cluster-wise cooperation distributes computational cost in a more balanced manner among clusterheads. In addition to reducing the dependence on a single entity of the global strategy, it also alleviates overhead imposed on almost every node by the link-wise approach.

The operation of CB module is depicted in Figure 6.2. The CB module consists

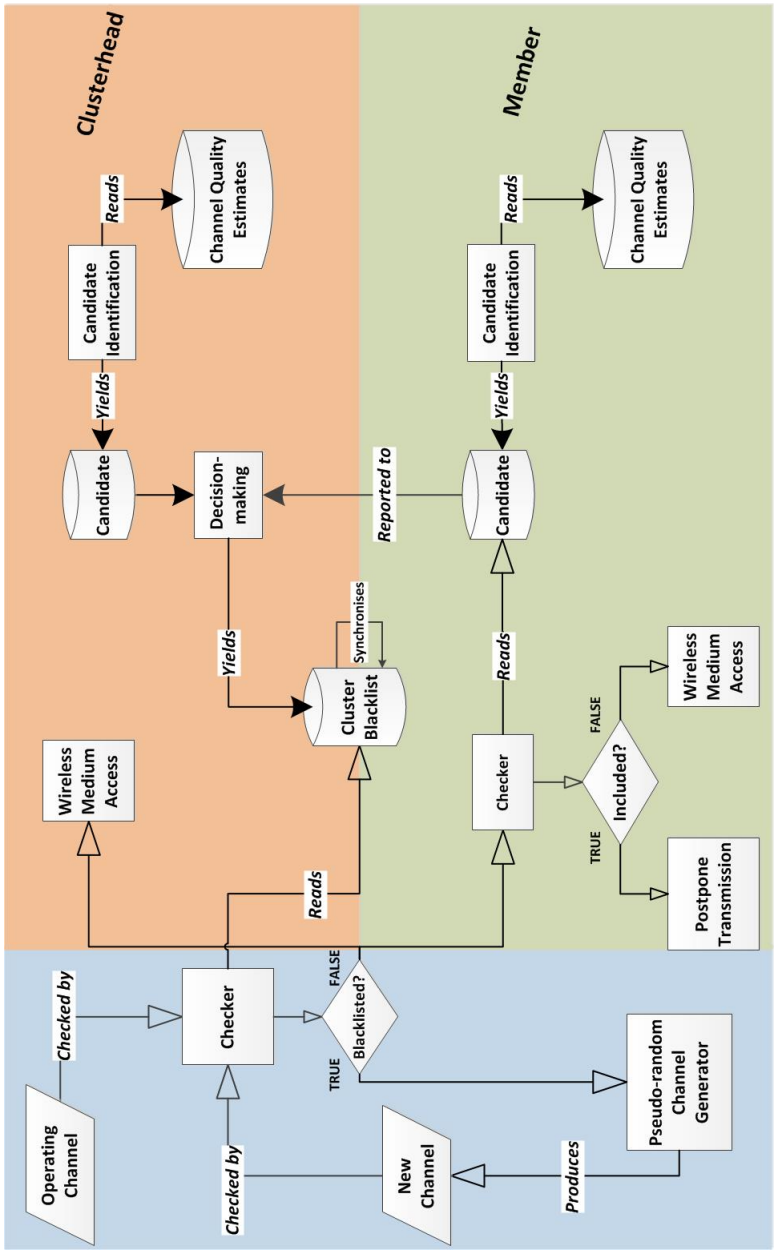


Figure 6.2: Operation of Cooperative Blacklisting. This module consists of *Cooperative Election* in NF slots and *Selective Channel Hopping* in Tx and Rx slots. Processes of these two logical components are denoted by solid and hollow arrows, respectively. Square boxes represent important function blocks; cylinders represent persistent information entities, whilst temporary data are shown in parallelograms. Clusterhead and members are represented in different colour blocks. The blue area to the left represents functions common to both node types.

of *Cooperative Blacklist Election* in NF slots and *Selective Channel Hopping* in Tx and Rx slots, which are denoted by solid and hollow arrows, respectively. The process differs between clusterhead and other members, which are in different colour blocks. Functions common to both node types are represented in the blue area in the left part of the figure. In the subsequent sections of this chapter, detail discussion of both *Cooperative Blacklist Election* and *Selective Channel Hopping* are provided.

6.3 Cooperative Blacklist Election

The cooperative election of blacklists is conducted in *Candidate Identification*, *Candidate Aggregation*, *Blacklist Decision-making* and *Synchronisation* phases. This section provides details of these phases shown in Figure 6.2, followed by discussions of invocation mechanisms for Cooperative Blacklist Election.

6.3.1 Candidate Identification

At the first stage of Cooperative Blacklist Election, nodes independently identify sets of undesirable frequencies by consulting their channel quality records produced by the Channel Quality Estimates (CQE) module. Based on findings in Section 6.1, undesirable channels are identified using certain noise threshold. These resulting channel collections represent tentative blacklists that individual nodes propose to use, which are hereafter referred to as *blacklist candidates* or simply *candidates*. No channels are actually excluded from the hopping sequence at this point since blacklist candidates are primarily intended to capture channel condition of different locations, which provides the basis for cooperation.

6.3.2 Candidate Aggregation

Blacklist candidates are reported to corresponding clusterheads in the Candidate Aggregation phase. To minimise extra energy consumption this may incur, candidate reports from member nodes are transmitted via piggyback on ordinary

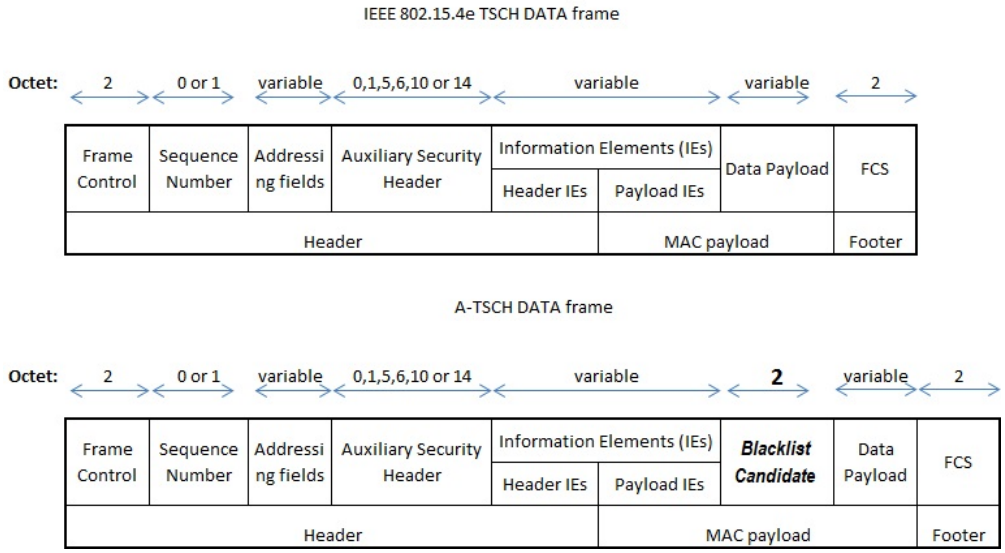


Figure 6.3: Format of A-TSCH DATA frames in comparison with that of IEEE 802.15.4e TSCH DATA frames.

data traffics. The format of data frame carrying candidate report is illustrated in Figure 6.3. An additional 2-bytes is reserved in the frame payload before the actual application data as highlighted in the figure. Each bit of this field corresponds to the local tentative decision for a specific channel.

Because A-TSCH operates in networks with tree-like topologies as described in Figure 3.3 of Section 3.2.2, all data traffics are forwarded by clusterheads except for those directly connected to the gateway node. Consequently piggybacked candidate reports are bound to reach their clusterheads. Two exceptional conditions may occur when certain node has no data to send or data are lost in transmission. In the former case, nodes with no data to transmit are not affected by risk caused by ambient noise hence the absence of their reports has minimal effect on overall performance. In the latter case, reports will eventually reach the clusterhead in re-transmission or subsequent data traffics.

6.3.3 Blacklist Decision-making

In the following Blacklist Decision-making phase, a clusterhead generates the decision on the blacklist for the cluster. The decision-making mechanism implemented in clusterheads is described in Algorithm 6.1. According to discussion in Section 2.3.3, the *hard combination* technique is adopted in this process since reported blacklist candidates contain binary ON / OFF decisions for specific channels, as opposed to the complete raw sensing data used in *soft fusion* [26]. Specifically, the algorithm employs a combination of AND-rule and OR-rule which are both common hard combination techniques [144, 26].

The algorithm can be invoked by the reception of candidate reports from member nodes. As show between line 3 and 8 in Algorithm 6.1, the clusterhead compares received candidate report with its existing record for this member node. If there is a mismatch, the election process is initiated by invoking the AND-rule which produces a list of channels unanimously identified as undesirable by all cluster members, as defined between line 15 and 27. Subsequently the OR-rule is applied to the clusterhead's own blacklist candidates and the list generated by the AND-rule (line 28 - 34). Consequently, any channel marked in either of the list is included in the blacklist for the entire cluster. The election can be alternatively triggered by an update of the clusterhead's own local candidates. In that case, only the OR-rule is invoked as the product of last AND-rule remains valid.

The algorithm is designed to provide suitable blacklisting accuracy without incurring excessive overhead. First, it recognises the fact that the clusterhead is at the centre of all communication within its cluster. Accordingly any candidate identified by the clusterhead is elected in the final cluster blacklist to ensure that the spectral condition of the clusterhead is reflected. Second, it incorporates information from cluster members to capture noisy channels that may negatively affect the members' side but undetected at the clusterhead's location. Third, it keeps the cooperation to a moderate level. Since spatially distributed members may experience different spectral condition, it is unrealistic to reflect

Algorithm 6.1 Cooperative Blacklist Decision-making

Data :

CSIZE : Number of cluster members
newCand[*m*] : Blacklist candidate reported by cluster member *m*
headCand : Local blacklist candidate generated by the clusterhead
memberCand[*CSIZE*] : Record array of blacklist candidates of cluster members
memberBlacklist : List of channels included in all *memberCand* entries

Result :

clusterBlacklist : The blacklist decided for the entire cluster

```
1: function MAIN
2:     while Device up and running do
3:         if newCand[m] received then
4:             if memberCand[m] ≠ newCand[m] then
5:                 memberCand[m] = newCand[m]
6:                 andRule( )
7:                 orRule( )
8:             end if
9:         end if
10:        if headCand updated then
11:            orRule( )
12:        end if
13:    end while
14: end function
```

Continued on the next page

Cooperative Blacklist Decision-making (continued)

```
15: procedure ANDRULE
        ▷ Only mark channels identified by all cluster members as undesirable
16:     for i ← 11, 26 do
17:         for j ← 1, CSIZE do
18:             if channel i not marked in memberCand[i] then
19:                 found = false
20:                 break
21:             end if
22:         end for
23:         if found ≠ false then
24:             Mark channel i in memberBlacklist
25:         end if
26:     end for
27: end procedure

28: procedure ORRULE
            ▷ Mark the channel if identified by the clusterhead or all members
29:     for i ← 11, 26 do
30:         if channel i marked in memberBlacklist OR in headCand then
31:             Mark channel i in clusterBlacklist
32:         end if
33:     end for
34: end procedure
```

all member candidates in the cluster blacklist. Consequently only channels that are commonly marked by all cluster members are included in the cluster blacklist.

There are limitations of using a uniform blacklist inside each cluster, since it is possible that the undesirability of certain channel is not universally associated with all member locations and thus omitted from the final elected blacklist. However, taking into account of the potential issues of link-wise blacklisting, this algorithm ultimately represents a choice made in favour of reliability over improvement. Additionally, this issue can be mitigated with adaptive channel access mechanism such as Algorithm 6.2 which will be described in Section 6.4.

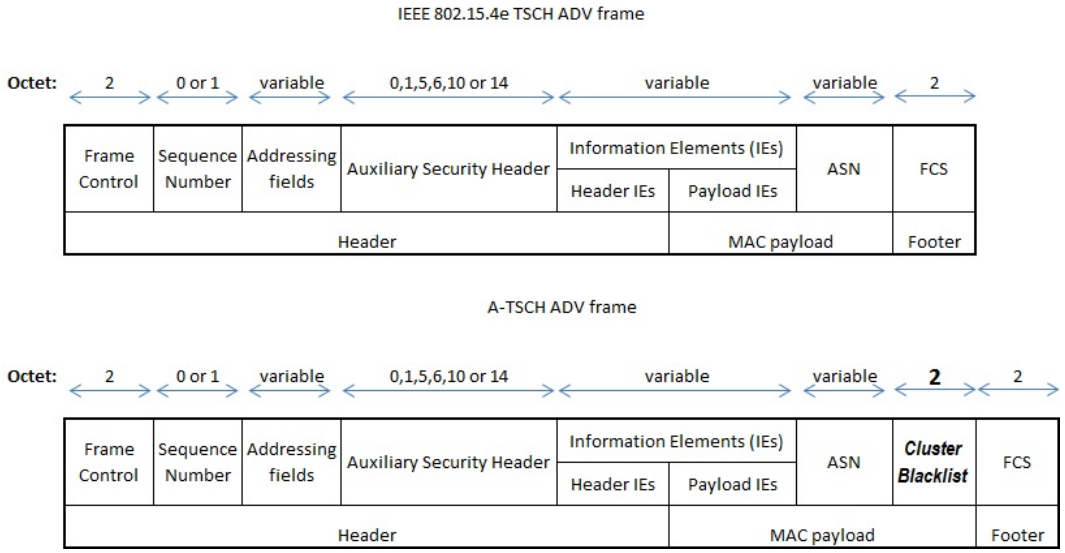


Figure 6.4: Format of A-TSCH advertisement (ADV) frames in comparison with that of IEEE 802.15.4e TSCH ADV frames.

6.3.4 Synchronisation

After the cluster blacklist has been decided by the clusterhead, the decisions need to be synchronised among cluster members in the last phase of Cooperative Blacklist Generation.

Similar to the measure adopted in Section 6.3.2, the dissemination of the cluster blacklist is integrated with the existing time synchronisation infrastructure using advertisement (ADV) messages. A 2-byte field is appended to the ASN information of the ADV frame as illustrated in Figure 6.4. ADV messages carrying both timing and blacklist information are placed into the outgoing buffer and are transmitted in the next ADV timeslot.

This arrangement provides a robust mechanism whilst minimises the communication cost and modification on existing architecture. Loss of ADV messages may only cause short delay in blacklist synchronisation as ADV messages are frequently exchanged to maintain the network timing. Additionally, blacklisting

is not activated in ADV timeslots which means problems in blacklist synchronisation does not affect the ability of the network to remain functional.

6.3.5 Invocation Mechanism

After the preceding discussions of cooperative blacklisting procedures, this section looks into the invocation mechanism of the module. The default duration of a timeslot implemented in OpenWSN is approximately 15 ms; accordingly a slotframe with ten slots accounts for less than 0.2 second. This is much shorter than the typical intervals between the data transmission of low-power wireless networks in IoT therefore it is neither suitable or necessary to invoke blacklist election in every NF slot.

The blacklist election can be invoked in *periodic* or *reactive* mode [33, 26, 121]. In the periodic mode, the election is regularly activated to maintain an up-to-date reflection of actual spectral condition. But choosing a suitable update period is often not straightforward. On the one hand, short update periods can result in excessively frequent changes of blacklist which can be counter-effective. The election ultimately relies on channel quality estimates to make sensible judgment, and suitable estimates can only be obtained with a reasonable amount of ambient noise readings. Overly brief invocation periods also incur more computation and transmission overhead, undermining network stability and increasing the risk of misaligned blacklist between communicating peers. On the other hand, long invocation periods, though preferable in terms of cost and stability, may cause blacklists to lag behind current spectral condition and result in inaccurate exclusion of channels. Since the spectral condition is environment and application specific, a feasible practice is to deduce proper invocation period based on trials conducted with real deployment. The performance of this periodic scheme will be tested in Chapter 7 with a number of different parameters.

In the reactive mode, blacklist elections are invoked as a reaction to certain

spectral event without definitive time constraints. There are a range of triggers that could be used for this approach. Nodes could actively monitor the success rate of transmission, and invoke a re-election of blacklist once the performance deteriorates to a certain level. Transmission success rates can be monitored in terms of either overall ETX or that of each channel individually [143]. The former comes with lower complexity but may be less sensitive if only a small number of channels are affected by noise, whilst the latter could be more responsive at the expense of increased computational overhead. A limitation of using transmission success rates as the trigger of blacklist election is that the notable rise of ETX is a type of *a posteriori* information, meaning the actual loss of packets must take place before remedial measures can be taken.

For the CB module of A-TSCH, an alternative dual-threshold criteria is devised for event-triggered invocation mechanism using noise thresholds identified in Section 4.2.5. Noise levels of -89 dBm and -90 dBm are defined as the upper and lower thresholds, respectively. Channels with noise floors excessive of the upper threshold are immediately selected for blacklist candidates and those below the lower threshold are considered desirable for transmission. Between these two thresholds is a buffer zone. Channels with noise floors falling into this region are thought to be in a transitional and unstable state and their status regarding blacklist candidate election are not changed. For example, if a channel's noise floor drops from -88.5 dBm to -89.5 dBm, it is unclear whether the decline is temporary or persistent. Therefore it is assumed to still possess the previously recognised quality. This arrangement of dual-threshold is designed to avoid excessively frequent candidate elections which lead to surge of communication and computational overhead. This event-triggered reactive invocation scheme is tested in experimental work presented in Chapter 7.

6.4 Selective Channel Hopping

The *Selective Channel Hopping* of CB module, active in Transmission (Tx) and Reception (Rx) timeslots, controls the channel hopping behaviour using blacklists generated by Cooperative Blacklist Election.

The operation of Selective Channel Hopping is described in Algorithm 6.2. After entering a Tx or Rx slot, an operating channel number is generated based on Equation (3.1). Instead of immediately communicating in this channel as with standard IEEE 802.15.4e, the cluster blacklist is queried to ascertain the validity of this channel. If the channel is not in the blacklist, the node proceeds to transmit or receive; otherwise a new channel that is admissible must be generated. To ensure the randomness of this new channel, a pseudo-random number generator unrelated to Equation (3.1) is adopted using ASN as the seed value. Since numbers produced by the generator are controlled by the seed and ASN is uniform within a network running A-TSCH, the new channel is generated in the pseudo-random yet coordinated way for communicating nodes.

In Rx slots, nodes proceed to data reception after a permissible channel for cluster blacklist is obtained. On the other hand, if the node is not the clusterhead and the slot type is Tx, the algorithm continues to check the channel against its own blacklist candidate. This is because a receiver in the topology of A-TSCH is the clusterhead for the nodes from which it receives data and the decision-making mechanism in Section ensures the channel desirability of the clusterhead is fully represented in the cluster blacklist. But channels undesirable for certain sender might not be included in the cluster blacklist and consequently additional measures are needed to avoid using these channels in transmission.

If a sender finds that the channel is included in its local blacklist candidate, the node gives up the chance to send in this Tx slot and wait for the next turn. This allows noisy channels at both ends of a wireless link to be excluded from

Algorithm 6.2 Selective Channel Hopping

Data :
clusterBlacklist : Cluster blacklist
localCand : Local blacklist candidate of non-clusterhead node
outQueue : Buffer for outgoing packets
ASN : ASN number

Result :
opChan : The operating channel to use for transmission or reception in this timeslot

```
1: function MAIN
2:     while (slotType == Tx) OR (slotType == Rx) do
3:         opChan = CHANNUMGEN(ASN)
4:                                          ▷ Get channel number using ASN
5:         while opChan is marked in clusterBlacklist do
6:             opChan = CHANNUMGEN(null)
7:                                          ▷ Get a new channel number
8:         end while
9:         if nodeType ≠ clusterhead then
10:            if opChan is marked in localCand AND outQueue is not full then
11:                Give up transmission and leave the packet in outQueue
12:            end if
13:        end if
14:    end while
15: end function

16: function CHANNUMGEN( param )
            ▷ Mark the channel if identified by the clusterhead or all members
17:    if param ≠ null then
18:        Return (parammod16) + 11
19:                                  ▷ Get channel number based on ASN
20:    else
21:        Return rand(11, 26)
22:              ▷ Get channel number with standard pseudo-random generator
23:    end if
24: end function
```

hopping sequence without necessitating pair-wise negotiation. Considering the fact that typical duration of a timeslot is about 150 ms, it should not notably affect the data. As an additional safeguard, this postponing measure is only taken when the outgoing queue has enough space for future packets. If the packet buffer is about to be overflown, data can be transmitted without checking the sender's blacklist candidate.

6.5 Summary

This chapter discusses the Cooperative Blacklisting module which is central to the A-TSCH framework. CB functionality mainly consists of Cooperative Blacklist Election and Selective Channel Hopping. Cooperative Blacklist Election is in charge of the production and synchronisation of cluster blacklists, which can be either activated periodically or triggered by certain events. The election of blacklists is achieved via cluster-wise blacklisting, which represents a trade-off between the global and link-wise strategies. With a modification of the DATA and ADV frame structures, cooperation is mainly performed with in parallel with the existing data transmission and timing synchronisation, incurring minimal operational overhead. Selective Channel Hopping regulates the utilisation of blacklist to exclude blacklisted channels from hopping sequence. A standard pseudo-random generator that is widely available in common programming libraries is employed to propose alternative channels once the default channel is found to have been blacklisted. Transmitting nodes additionally check their own candidates to decide whether to postpone the transmission in order to avoid using undesirable channels that are only known to the sending side. Consequently, wireless medium is only accessed in permissible channels.

Chapter 7

A-TSCH Performance Evaluation

This chapter presents an investigation into the performance of A-TSCH through experiments. Firstly, Section 7.1 specifies the settings of experiments. Then the criteria for performance assessment are introduced in Section 7.2, followed by analysis of experimental results in Section 7.3. A summary of experimental findings is provided in Section 7.4.

7.1 Experiment Settings

This section describes experiments conducted to evaluate the performance of A-TSCH. The experimental environments and the deployment of devices are introduced at first, and then different configurations for A-TSCH implementation are specified.

7.1.1 Environment and Deployment

Experiments were conducted at Location A introduced in Section 4.1 because this workplace environment provides the most dynamic noise profile out of the three available options. First, the overall detectable noise strength at Location A is more suitable for assessing the effectiveness of A-TSCH, compared with Location

B and C where ambient noise is too weak to exert significant influence on communication as discussed in Chapter 4. Second, the temporal variation of noise floor in terms of the magnitude and distribution across the 2.4 GHz band is able to provide a wide scope of different spectral condition, therefore more interesting measurements of performance can be obtained at this location. Third, the relative spaciousness of this workplace makes it possible to capture certain level of spatial diversity with increased size of deployment.

A total of 18 GINA motes [13] were used for the experiments. Note that the hardware was chosen in the interest of convenience and A-TSCH is also compatible with other platforms such as TMote Sky [14]. The placement of nodes is illustrated in Figure 7.1. Node *1*, shown in the brown circle, was attached to a desktop and appointed as the gateway. Positions of remaining motes are shown as blue circles with corresponding ID.

7.1.2 Configuration

The implementation of A-TSCH is developed on top of UC Berkeley's OpenWSN stack [145]. Because 6LoWPAN is an integral part of the OpenWSN stack, JANET IPv6 tunnelling service is installed and activated in the desktop to which the gateway node is attached. Accordingly sensing data generated inside the low-power network can be accessed over the Internet.

All motes are programmed to run A-TSCH and the structure of the slotframe used in our implementation is depicted in Figure 7.2. The length of slotframe is set to nine. The first timeslot is ADV for network synchronisation, followed by five TX/RX slots for the exchange of data packets. The two subsequent NF timeslots are reserved for noise floor detection. Within each NF slot, 128 noise samples are collected and averaged to produce the noise floor reading for the specific slot. The last slot which is not assigned to any task.

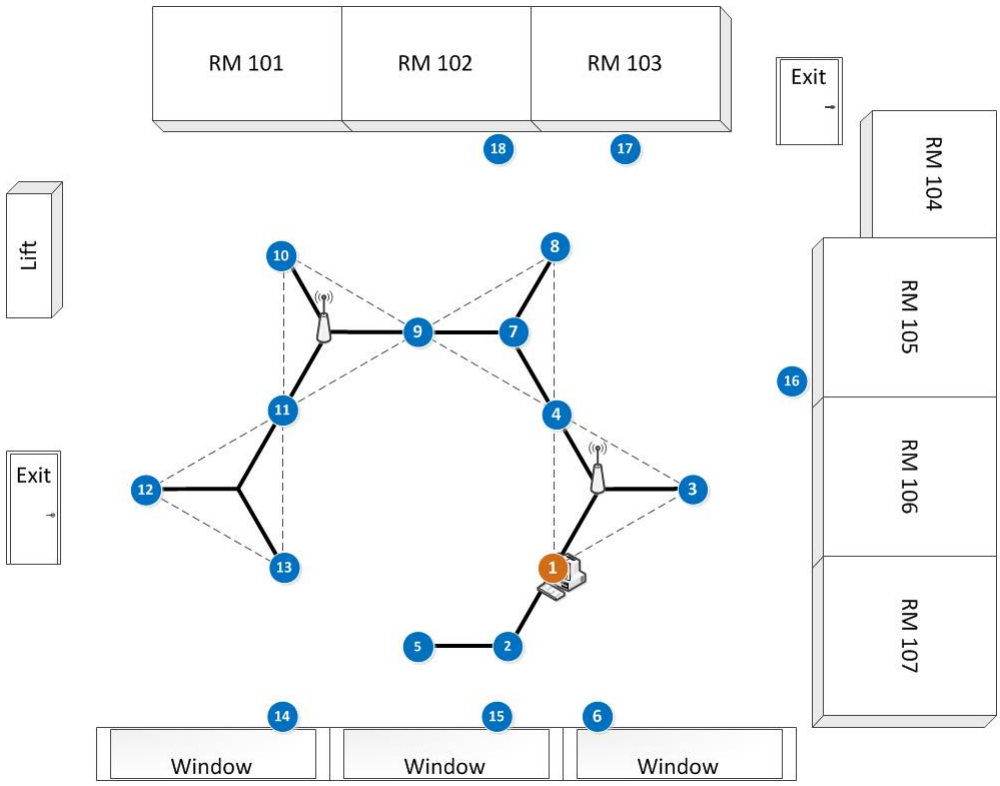


Figure 7.1: The deployment of GINA motes at Location A. Circles denote mote positions with corresponding ID. The brown circle represents the gateway node and the remaining motes are denoted by blue circles.

This arrangement is a practical choice taking into account the balance between transmission efficiency and detection accuracy. Because blacklists are not applied in NF slots, channels selected for noise detection follow the pattern highlighted in the NF columns of Table 7.1 which shows the default channel hopping sequence without blacklisting. The red cells collectively constitute the entirety of IEEE 802.15.4 frequencies, therefore it takes eight slotframes to sample the noise floors of all channels. The slot time in our implementation is 15 milliseconds hence eight slotframes take 1080 milliseconds, that is, slightly over one second.

Experimental configurations shown in Table 7.2 are used to investigate A-TSCH performance in distinct scenarios.

Slot Offset	0	1	2	3	4	5	6	7	8
Slot Type	ADV	Tx / Rx	Tx / Rx	Tx / Rx	Tx / Rx	Tx / Rx	NF	NF	Idle

Figure 7.2: A-TSCH slotframe used in experiments.

Table 7.1: Default Channel Hopping Pattern for Slotframe of nine timeslots. Values in the table are IEEE 802.15.4 channel numbers. A complete spectrum sensing cycle of 16 channels in the NF slots is highlighted in red.

Slot Type / Slotframe	ADV	Tx Rx	Tx Rx	Tx Rx	Tx Rx	Tx Rx	NF	NF	IDLE
1	26	11	12	13	14	15	16	17	18
2	19	20	21	22	23	24	25	26	11
3	12	13	14	15	16	17	18	19	20
4	21	22	23	24	25	26	11	12	13
5	14	15	16	17	18	19	20	21	22
6	23	24	25	26	11	12	13	14	15
7	16	17	18	19	20	21	22	23	24
8	25	26	11	12	13	14	15	16	17
9	18	19	20	21	22	23	24	25	26

Table 7.2: Configuration of experiments that investigate the performance of A-TSCH. Experiments were conducted using five different combinations of settings, and are hereby identified as Experiment I through V.

Experiment ID	**I**	**II**	**III**	**IV**	**V**
Blacklisting Invocation	N/A	Periodic			Event-triggered
Invocation Period (minute)		1	2	5	N/A
Noise Threshold		-89 dBm			Dual-theshold
Estimator		ES, KF, KFES			
Packet Size (byte)	128				
Number of Packets	600				
Transmission Rate	1 packet every 2 seconds				
Iterations	9	3 per (Period,Estimator)			9 per Estimator
Total Runs	63				

Experiment I has blacklisting disabled to represent standard IEEE 802.15.4e TSCH. In Experiment II to V, the performance of A-TSCH is tested using invocation mechanisms discussed in Section 6.3.5. Specifically, periodic invocation is used in Experiment II to IV with using the noise threshold of -89 dBm and periods of one, two and five minutes, respectively. In Experiment V, the event-triggered mechanism is deployed with the dual-threshold criteria introduced in Section 6.3.5.

The ES, KF and KFES estimators discussed in Chapter 5 are also tested. In Experiment II to IV, every possible combination of invocation period and estimator is tested, resulting in nine iterations each. Because invocation periods do not apply in Experiment V, nine iterations using a specific estimator are carried out. Accordingly, a total of 63 rounds of test took place. In each test, motes are configured to send 600 packets to the gateway at 0.5 Hz.

7.2 Evaluation Criteria

The performance of A-TSCH is evaluated in both control plane and data plane. On the one hand, the control plane investigation ascertains whether the blacklisting operation of A-TSCH behaves correctly with respect to the spectral condition. On the other, the data plane evaluation demonstrates the effect of A-TSCH on data transmission.

7.2.1 Control Plane Metrics

The correctness of blacklisting operation is an important evaluation criterion since the performance of A-TSCH is closely related to its hopping sequence. For this purpose, the metric of *Blacklisting Rate* (BR) is introduced. As defined in Equation (7.1), $BR[c]$ reflects the probability for channel c to be blacklisted during the experiments. Accordingly it quantifies how frequently certain channel was recognized as noisy and excluded by the blacklisting process.

$$BR[c] = \frac{Number\ of\ blacklisting\ decisions\ that\ include\ channel\ c}{Total\ number\ of\ blacklist\ decisions\ made} \qquad (7.1)$$
$$c \in [11, 26]$$

7.2.2 Data Plane Metric

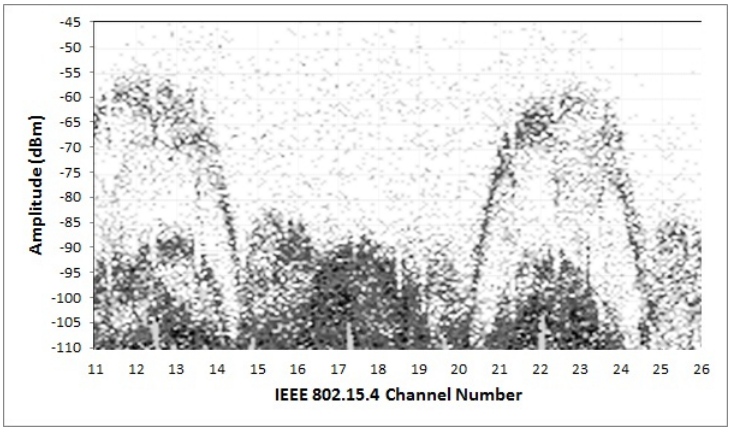
The most significant indicator of performance is the improvement of transmission success rates, as it is the very motivation for A-TSCH to be devised. According to the definition introduced in Section 4.2.2, Expected Transmission Count (ETX) is calculated based on numbers of transmitted and received packets to indicate the reliability of packet delivery.

7.3 Results

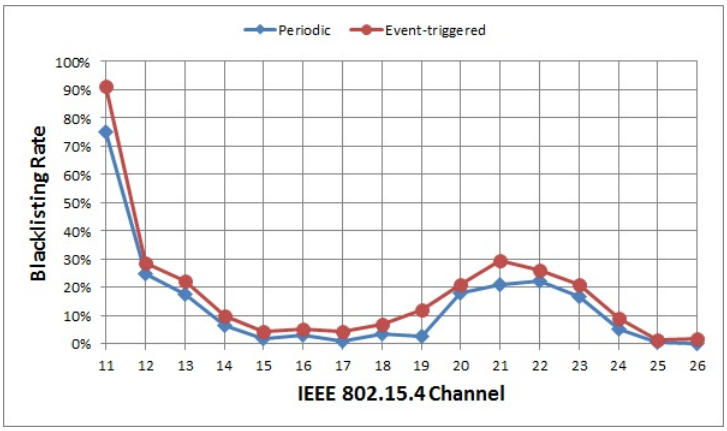
7.3.1 Control Plane

Since the performance of A-TSCH is largely attributed to its blacklisting operations, the blacklist updates during the experiments are examined to ascertain whether A-TSCH is able to correctly exclude channels in accord with noise floor conditions. For this purpose, the channel condition was obtained using Wi-Spy, a commercial spectrum analyser [146] to provide trusted knowledge of noise floor and the resulting traces of noise density are presented in Figure 7.3a.

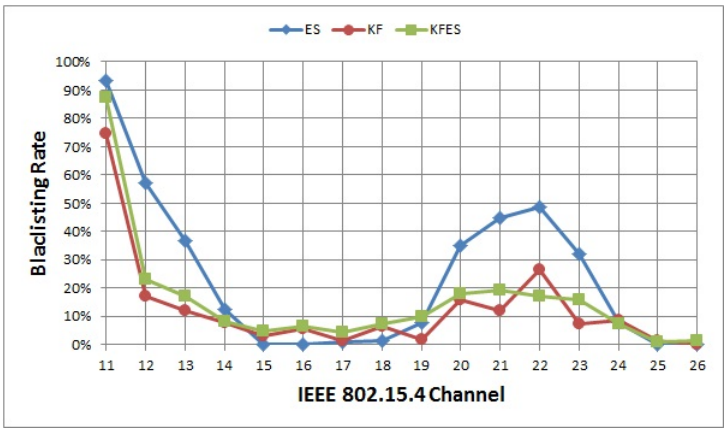
Curves in Figure 7.3b represent average channel BR using different invocation modes which are found to be in agreement with the actual noise condition. Figure 7.3a, 11 to 13 and 21 to 23 are the most frequently blacklisted channels, whilst the relatively quiet channel 25 and 26 are the least blacklisted ones. This demonstrates that A-TSCH is able to correctly identify and blacklist the most noisy channels, leading to enhanced transmission reliability. It is also observed that the BR of the triggered mode are slightly greater than that of the periodic version. This is because blacklist elections are only activated when there is some



(a)



(b)



(c)

Figure 7.3: The noise floor density is obtained with WiSpy and presented in (a). The average blacklisting rates (BRs) are plotted in (b) for different invocation modes and in (c) for different A-TSCH estimators.

channels to be excluded in event-triggered mode, whilst it is not necessarily the case for scheduled elections in periodic mode.

In order to look at the behaviours of different channel quality estimators, the average BRs of ES, KF and KFES estimators are plotted in Figure 7.3c. It is observed that the BR in general is higher with ES than the other two. This again demonstrates that ES is more apt to react to short-term changes in noise condition, which may be an advantage in terms of the capture of blacklisting opportunities and a disadvantage if the the estimate it generates vary too wildly. This also partly explains ES's good performance in event-triggered mode as illustrated in Figure 7.4b and Table 7.4. Furthermore, KFES appears to yield BR that are at a level between ES and KF, which is consistent with the earlier observations on ETX performance.

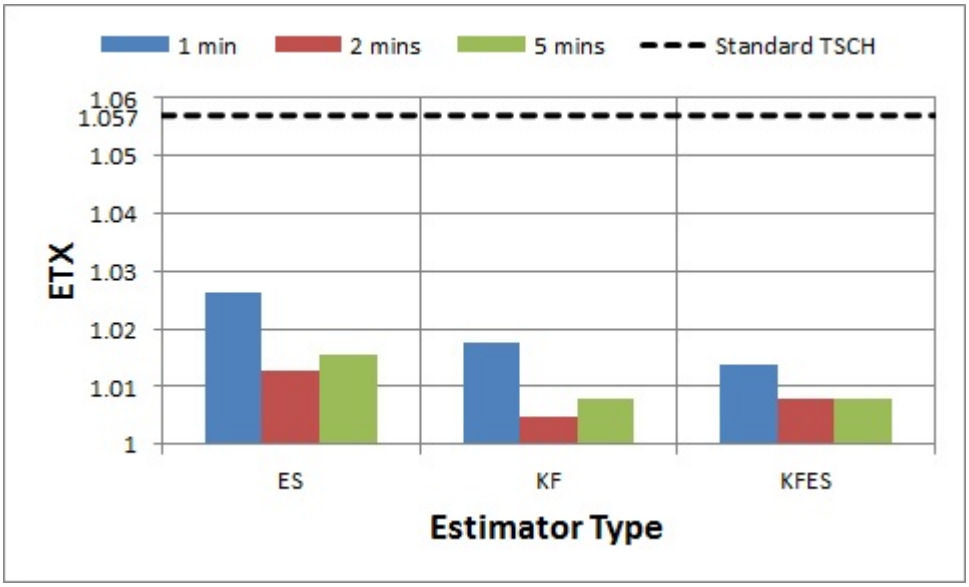
In both Figure 7.3b and 7.3c, there is a high BR for channel 11. Although this does not contradict Figure 7.3a, the magnitude at this channel is slightly disproportionate compared with the level of BR for almost equally noisy channels such as 21 and 22. This observation may be due to the fact that there is only one WiSpy channel analyser and it has to be attached to a fixed desktop. Therefore its readings are most accurate at that position. The motes, on the other hand, are spread over the environment and may be exposed to noise events that are either not detected by Wispy or insufficient to be reflected in the graph. This once again indicates that noise profiles are closely associated with specific environments and further investigation into estimation techniques should be conducted in future work. In spite of the diversity discussed above, the shapes of BR lines in Figure 7.3b and 7.3c are all in agreement with the profile of noise floor densities in Figure 7.3a, demonstrating the overall correctness of A-TSCH estimator and the blacklisting operation.

7.3.2 Data Plane

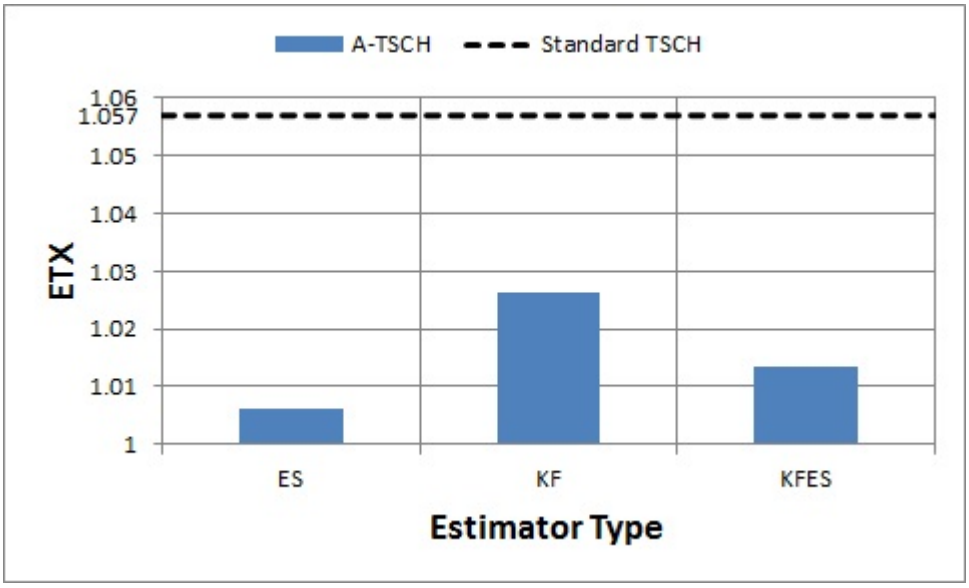
To facilitate the analysis, we in this section focus on data transmitted from Node *2*, *4*, *5*, *9*, *11*, and *13*, since they cover the key positions of the hexagonal area in the environment. The average ETX achieved with different estimators are computed and reported in Figure 7.4, where the results of standard TSCH are computed and provided as a dotted baseline.

The performance of A-TSCH with periodic invocation scheme are illustrated in Figure 7.4a. The resulting reduction in ETX is also listed in Table 7.3. Three individual segments of the figure denote results achieved with different estimators respectively indicated on the horizontal axis, and inside each segment the ETX of different invocation periods are represented by three columns. It is observed that ETX is visibly higher with the period of 1 minute than the other two, at least 0.8% less in average ETX reduction according to Table 7.3. This reflects the potential counter-effectiveness of unnecessarily frequent blacklist election. The shorter intervals between each election, the greater chance the resulting cluster blacklist and candidates are affected by short-term fluctuation of noise floors, which increase the probability of suboptimal blacklisting. Moreover, exceedingly frequent election also increase the risk of desychronised cluster blacklists. If member nodes do not learn the updated cluster blacklist in time, their may end up transmitting in channels already excluded by their clusterhead. The periods of two and five minutes yield comparable ETX, and no definitive findings can be drawn based on existing evidence.

Variation in ETX reduction is also existent between A-TSCH estimators, which is visualised in a comparison across all segments in Figure 7.4a and can be found in Table 7.3. Based on data in the rightmost column of Table 7.3, KF and KFES are the best-performing estimators, indicating their ability to yield accurate estimation of the desirability of frequencies. Also it is observed that the level of ETX reduction using ES estimator varies more notably with different periods, reflecting its relatively less stable feature.



(a)



(b)

Figure 7.4: The Expected Transmission Count (ETX) observed with distinct experiment settings. The horizontal axis is the type of estimator and the vertical axis denotes the average ETX. The results of Experiment II, III and IV for A-TSCH using periodic invocation mechanism are illustrated in (a). Inside each segment of corresponding estimator type, 3 bars denote different choices of invocation period. In (b) the results of Experiment V for event-triggered invocation scheme are depicted. In both (a) and (b) the ETX achieved in Experiment I with standard TSCH is provided as a baseline.

Table 7.3: The ETX reduction over standard TSCH achieved with A-TSCH using periodic invocation. Configurations correspond to Experiment II, III and IV. Percentages are calculated with respect to different experiment settings. Averages for individual estimators and periods are provided in the rightmost column and last row, respectively.

Invocation Period / Estimator	1 min	2 min	5 min	Average
ES	2.917%	4.194%	3.910%	3.674%
KF	3.721%	4.935%	4.636%	4.431%
KFES	4.084%	4.636%	4.652%	4.457%
Average	3.574%	4.588%	4.399%	

Table 7.4: The ETX reduction over standard TSCH achieved with A-TSCH using event-triggered invocation. Configurations correspond to Experiment IV. Percentages are calculated with respect to different experiment settings. Averages for individual estimators are provided in the rightmost column.

Estimator	ES	KF	KFES	Average
ETX Reduction	4.793%	2.901%	4.115%	3.937%

The results of event-triggered invocation scheme as described in Section 6.3.5 are illustrated in Figure 7.4b and Table 7.4. Unlike previous experiments, ES estimator in this mode outperforms the other two. A possible explanation is that the simplicity of ES enables it to capture changes in noise floor more quickly therefore more suitable for detecting short-lived events. As a result it is desirable for effectively exploiting blacklisting opportunities. Kalman filter, in comparison, tend to evolve gradually to optimal estimates hence its effect is less notable. K-FES, combining the straightforwardness of ES and stability of KF, provides the

moderate ETX reduction.

It should be noted in Figure 7.4 and quantification in Table 7.3 and 7.4 that A-TSCH uniformly deliver lower ETX than standard TSCH regardless of specific settings. This proves the enhancing effect of A-TSCH on transmission success rates. Furthermore, A-TSCH based on ES and KFES were observed to provide comparable performance with either periodic and event-triggered invocation schemes. But event-triggered blacklist is considered superior to the periodic scheme as its flexibility facilitates adaptation to changing environment and noise profiles which are essentially unpredictable. Accordingly we prefer the event-triggered approach and use this invocation scheme for experiment in the next chapter.

7.4 Summary

In this chapter we evaluated A-TSCH framework in terms of its ability to correctly identify suitable blacklist and enhancing effect on transmission success rates. Experiments were conducted in the open workplace environment known as Location A throughout this work. The blacklisting decisions were examined utilising a WiSpy channel analyser. We found that A-TSCH was able to correctly make blacklisting decisions in accord with the spectral condition. ETX were also computed with respect to different configurations including invocation mode and estimator types. The most significant overall ETX reductions have been observed with KF and KFES in the periodic mode and ES and KFES in the event-triggered mode. Despite the fact no deterministic verdict has been worked out on different configurations, A-TSCH uniformly delivered better transmission success rates than standard TSCH.

Chapter 8

Cross-layer Enhancement

As pointed out in Chapter 1, an important ingredient of adaptive communication for IoT is the cross-layer optimisation [121]. This chapter investigates the synergy between MAC layer A-TSCH and an IP-based routing protocol at the network layer. In Section 8.1, the routing protocol for low power wireless networks is reviewed. Then the spectrum-related routing metric for augmented RPL is introduced in Section 8.2. The effectiveness of the new metric is first examined through simulations presented in Section 8.3. Subsequently experiments are conducted to test its performance in real deployment. The settings of experimental work are specified in Section 8.4, and evaluation criteria are described in Section 8.5. Collected experimental results are discussed in Section 8.6. And at last our findings are summarised in Section 8.7.

8.1 Routing in Low-Power Wireless Networks

This chapter specifically focuses on the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) developed by IETF Routing Over Low power and Lossy networks (ROLL) working group. Low-power wireless networks are therefore required to implement IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) above the MAC layer. As discussed in Chapter 2, 6LoWPAN reduces the transmission overhead of standard IPv6 through header compression,

providing a lightweight solution to the issue of interoperability and integration for existing IP-based Internet [111, 113, 121]. In this section we present a brief review of some key facts of RPL.

RPL is an IP routing protocol designed to accommodate the requirements of low-power wireless networks. Typical characteristics of such networks include: support for only low data rate, unstable transmission reliability, restrained processing power, memory and energy of router nodes, and so forth. Additionally, low-power wireless networks consist of potentially large number of nodes and the traffics are often Multipoint-to-Point (MP2P) or Point-to-Pultipoint (P2MP) rather than Point-to-Point.

In order to address the issue that low-power wireless networks typically do not have predefined or static topologies, RPL perceives a network topology as a Destination Oriented Directed Acyclic Graph (DODAG), which has a cycle-free structure where graph edges lead to common data sink, also referred to as the DODAG root and is in many cases the gateway node [114, 115]. As a distance-vector protocol, each node in RPL DODAG is assigned a *Rank* value which indicates its position in the topology relative to the root: the higher a Rank, the more distant the node from the gateway. Consequently, the routing of packets is essentially the process of choosing the neighbour with the lowest Rank at each hop in typical scenarios where data are aggregated at the gateway.

The DODAG is constructed and maintained in RPL based on Object Function (OF) and periodic exchange of DODAG Information Object (DIO) messages [119]. An OF defines the rule of Rank calculation and also the selection of the next-hop (also referred to as *parent*). The construction of a DODAG starts when the root advertise its presence using DIO messages. Nodes that receive the DIO make decision on whether or not to join the DODAG grounded at this specific root based on their OF settings. Once they agree to join, they will recognise the root as their routing parent. Accordingly these nodes now form the second

tier of the graph and their own Ranks are calculated as additive to the original Rank of their parent. Nodes that are configured to act as router will repeat the above process by broadcasting their Ranks to their neighbours so that subsequent tiers are also established. Nodes that are not configured to act as router or not selected by any of its neighbours as their parent are leaf nodes. The construction of DODAG is completed once all nodes at the lowest tier are leaf nodes.

A multitude of metrics can be defined in OF for the computation of Rank values. As suggested in [118], metrics are in the broad categories of node and link status. Node status may be workload, residual energy level, hop count; and link status may refer to throughput, latency or reliability. The version of RPL implemented in the OpenWSN protocol stack [145] uses ETX records of communication with individual neighbours as the default routing metric for selecting next-hops.

8.2 Spectral Routing Metric for RPL

In this section, we propose a new routing metric for RPL which enables underlying spectral condition to be taken into account in choosing data paths. The aim and objectives of the proposal are described, followed by formal definitions of the metric and its components.

8.2.1 Aim and Objectives

Due to the intrinsic unreliability of wireless medium, the spectral condition bears significant effect on the quality and cost of certain links. Existing versions of RPL monitor changes in links quality primarily using ETX records. However, ETX only indirectly reflet the effect of spectral condition and may be slow in capturing updated situations as notable changes in ETX values often occurs after a series of packets have been already lost. Our new spectral routing metric is proposed in order to enable more suitable and timely routing decisions to be made for low

power wireless networks in the presence of ambient noise.

To this end, the metric needs to be able to capture the spectral conditions of the current node and potential next-hops. The candidate blacklists in A-TSCH framework are generated based on ambient noises of IEEE 802.15.4 channels in the 2.4 GHz band observed at individual nodes, and are therefore used as the source of spectral information.

In the context of low-power wireless networks using A-TSCH, the spectrum-related costs associated with certain links mainly come from the difference between blacklist candidates independently yielded by communicating nodes. According to the description of the cooperative blacklisting decision-making in Section 6.4, only channels universally identified by all cluster members are included in the cluster blacklist. If a channel generated in pseudo-random channel hopping is only included in the local blacklist candidates, the sending node postpones the transmission and wait for the next turn. As the difference between the blacklist candidates grows, the chance for this delay in transmission increases which would lead to the point where sender has not choice but to carry out transmission in locally undesirable channels because the outgoing queue is being overflown. Consequently, the disparity between blacklist candidates should be captured in our spectral routing metric.

Blacklist candidates also provide other important information of link costs in addition to their disparity. As previously discussed, the communication should ideally take place in channels that are not present in blacklist candidates of either ends of the link. Therefore the number of commonly desirable IEEE 802.15.4 channels is another indication of the costs of links to potential next-hops. As a simplistic example which shows the importance of this indicator, consider the difference between the two links that have identical blacklist candidates with size of 1 and 15. Whilst these two links apparently have the same level of blacklist disparity, the former has a total of 15 usable channels in hopping sequence whereas

the latter leaves only one channel available.

8.2.2 Metric Definitions

We term the new routing metric as *Spectral Link Cost* (SLC), which is a composite scalar value jointly determined by the *Disparity* (DI) and *Exhaustion* (EH) factors that capture the two types of blacklist-related information discussed previously. In the following paragraphs, the definitions of DI and EH are provided first, and then the metric of SLC is formally introduced. To facilitate our discussion, equations in subsequent discussions are presented with respect to the hypothetical link connecting Node P and Node Q. The blacklist candidates of these two nodes are denoted by BC_P and BC_Q, respectively. Additionally, the 16 IEEE 802.15.4 frequencies in 2.4 GHz band are collectively represented as $\mathbb{U}$, thus its set cardinality $|\mathbb{U}|$ equals 16.

8.2.2.1 Disparity (DI) Factor

The DI factor measures the level of difference between the blacklist candidates of two ends of certain link. As defined in Equation (8.1), DI_{P_Q} takes the value of zero when the union of both BC_P and BC_Q is empty, since no spectrum-related cost exist when none of the channels are actually blacklisted. Otherwise, DI is calculated based on the ratio between the number of IEEE 802.15.4 channels commonly identified in BC_P and BC_Q, and those collectively included in the union of both candidates.

$$DI_{P\_Q} = \begin{cases} 0, & \text{if } BC_P \cup BC_Q = \varnothing. \\ \dfrac{2 \cdot |BC_P \cup BC_Q| - |BC_P \cap BC_Q|}{|BC_P \cup BC_Q|} = & 2 - \dfrac{|BC_P \cap BC_Q|}{|BC_P \cup BC_Q|}, \text{otherwise} \end{cases} \tag{8.1}$$

From probability theories we know that $(BC_P \cap BC_Q) \subseteq (BC_P \cup BC_Q)$, therefore $0 \leq \frac{|BC_P \cap BC_Q|}{|BC_P \cup BC_Q|} \leq 1$ is within the range between [0,1] provided that $BC_P \cup BC_Q \neq \varnothing$. The ratio takes the maximum value of 1 when Node P and Q have identical blacklist candidates, that is $|BC_P \cup BC_Q| = |BC_P \cap BC_Q|$. As the difference between BC_P and BC_Q begins to arise and increase, $\frac{|BC_P \cap BC_Q|}{|BC_P \cup BC_Q|}$ decreases and reaches the minimum value of 0 when BC_P and BC_Q become so different that they contain no IEEE 802.15.4 channels in common. During this process, the value of DI_{P_Q} changes from 1 to 2 according to Equation (8.1). Therefore the spectral cost caused the disparity between blacklist candidates of certain link represented by the DI is within the ranges $[1, 2]$, except the case of both empty blacklist candidates where DI equals 0.

8.2.2.2 Exhaustion (EH) Factor

The EH factor measures the potential overhead incurred by the exhaustion of mutually desirable channels for both sending and receiving sides of the link, that is, the total of channels collectively included in both blacklist candidates.

$$EH_{A\_B} = \begin{cases} 255, & \text{if } BC_P \cup BC_Q = \mathbb{U}. \\ \\ EH_{A\_B} = \dfrac{|\mathbb{U}|}{|\mathbb{U}| - |BC_A \cup BC_B|}, & \text{otherwise} \end{cases} \tag{8.2}$$

The definition of DH_{P_Q} is provided in Equation (8.2). When $|BC_P \cup BC_Q|$ equals $\mathbb{U}$ which is the complete set of IEEE 802.15.4 channels, no mutually desirable channels exist on the link, therefore the cost represented by EH is assigned to the maximum value of 255, which is the largest possible for 8-bit field used in this work. Otherwise, EH ranges in $[1, 16]$ corresponding to the number of channels included in either of the blacklist candidates.

8.2.2.3 SLC

The definition of the metric of SLC is provided in Equation (8.3) combining the DI and EH factors. Two special cases are considered: according to the previous definitions of DI, no disparity cost is incurred when $BC_P \cup BC_Q = \varnothing$. In fact, this applies not only to DI but also to the entire SLC because spectral cost is irrelevant to A-TSCH when no channels are blacklisted, therefore SLC equals 0 in this case. Additionally, the EH factor takes the maximum value when $BC_P \cup BC_Q = \mathbb{U}$ for that no mutually desirable channels are existent. Similarly, this also applies to the overall spectral link cost and therefore SLC equals 255 under this condition. Apart from these two special situations, SLC_{P_Q} is calculated as the product of DI_{P_Q} and EH_{P_Q}, which is in the range of $[1, 32]$. Links with lower SLC are preferred over those of higher SLC as it implies smaller overhead associated with A-TSCH and less likely disruption at MAC layer.

$$SLC_{P\_Q} = \begin{cases} 0, & \text{if } BC_P \cup BC_Q = \varnothing. \\ 255, & \text{else if } BC_P \cup BC_Q = \mathbb{U} \\ DI_{P\_Q} \cdot EH_{P\_Q}, & \text{otherwise} \end{cases} \tag{8.3}$$

In our version of RPL, SLC is used as the main routing metric and the default ETX-based metric implemented in OpenWSN is employed as the tie-breaker according to the recommendation in [118]. This arrangement is also motivated by the concern that ETX can provide useful information when factors other than ambient noise such as multipath fading have dominant effect. Note the fundamental operation of RPL remains intact and the modification is only made to the specific criteria used when making routing decisions.

8.3 Simulation

The effect of our new routing metric was investigated in software simulation before hardware implementation. In this section we first provide a brief introduction to the simulator and related tools. Subsequently the configuration of simulation is specified. And finally the results is presented to provide preliminary evidence of the performance of SLC.

8.3.1 Contiki and Cooja Simulator

The Contiki operating system and the Cooja simulator are used for the work. Contiki is an open source operating system for devices with low-power microcontrollers in the Internet of Things [147]. Based on the concept of protothreads which are stackless threads library built on C macros, Contiki provides a highly portable, low RAM overhead, event-driven protocol stack for small embedded systems. In particular interests of this work, Contiki comes with fully certified IPv6 stack (contributed by Cisco), 6LoWPAN adaptation layer, and an implementation of RPL called ContikiRPL. Contiki also supports a range of hardware platforms such as Tmote Sky, MicaZ, Wismote and so forth.

Cooja is a Java-based simulator for Contiki. Systems are still developed and compiled in Contiki; but instead of being downloaded to actual hardware, they are executed in Cooja which simulates the behaviour of supported mote types and environmental events.

8.3.2 Configuration

Different configurations were used for simulation and combinations of parameter settings were as summarised in Table 8.1. Networks were modelled with different number of motes in areas of different sizes. The transmission range of motes also differ accordingly. Motes were configured to transmit to the sink node at random intervals between 1 and 60 seconds. Every simulation run lasted for five

simulation-hours at the speed rates lower than 500%.

Table 8.1: Different combinations of parameter settings used in Cooja simulation.

Parameter	**Settings**			
Network size	100m * 100m			
Number of motes	100	200	300	500
Transmission range	10m	5m, 10m		
Transmission interval	Random between 1 and 60 seconds			
Duration	5 simulation hours			

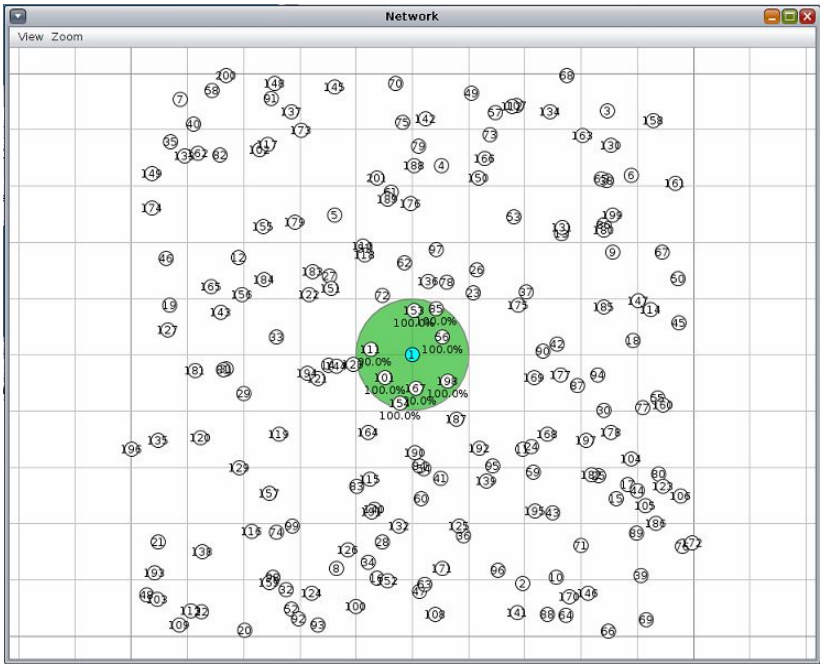


Figure 8.1: An example of simulated network in Cooja. 200 motes were randomly placed in an area of 100m * 100m. Motes were made to send packets to the sink located at the centre at random intervals between 1 to 60 seconds. The green circle depicts the 10-meters mote transmission range used in this example.

Because we intended to focus on the effect of routing metrics, low level environmental events such as channel noise levels were modelled directly through the probability of packet loss at the MAC layer. And changes in ambient noise were mimicked by making random association between channels and packet loss factors. Simulation using certain parameter setting was assigned with a unique random seed and carried out twice using SLC and ETX as the main RPL metric, respectively. Consequently factors such the shifting of ambient noise profile and transmission interval were identical for this pair of simulations.

An example of a simulated low-power network is shown in Figure 8.1. Each grid in the figure accounts for a 10m * 10m area and motes were randomly distributed within 100m * 100m. In this simulation we modelled the MP2P scenario where these 200 motes send data to the data sink (Node *1*) located at the centre. The transmission range was set to 10 meters and the green circle depicts the transmission radius covered by the sink.

8.3.3 Results

The simulation results for individual configurations are presented in Table 8.2. The first two columns specify the number of motes and transmission ranges, respectively. Transmission performance for ETX-based and SLC-based RPL metrics is averaged and provided in the last two columns. The results are also visualised in Figure 8.2.

It can be found that in all configurations RPL using SLC as the main routing metric uniformly delivers superior transmission performance than RPL using ETX-based metric. The resulting overall distinction of all simulation configurations is shown in the last row. These differences translated into Packet Delivery Ratio (PDR) amount to more than 3% improvement in the transmission success rates as a result of using SLC for RPL. It is observed that the curves of SLC sometimes take a dip with larger network density. A possible explanation is the

Table 8.2: Simulation results for different routing metrics using different configurations

Mote Number	Transmission Range	Performance (ETX)	
		SLC-based RPL	ETX-based RPL
100	**10m**	1.087	1.105
200	**5m**	1.089	1.116
	10m	1.103	1.124
300	**5m**	1.076	1.122
	10m	1.087	1.136
500	**5m**	1.106	1.146
	10m	1.086	1.178
Average		1.090	1.132

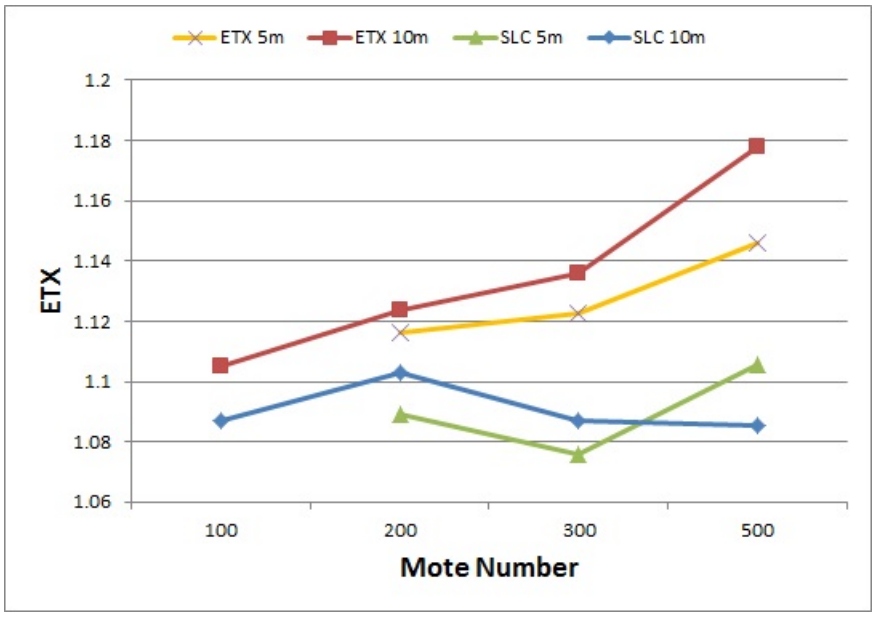


Figure 8.2: Transmission performance for ETX-based and SLC-based RPL metric in Cooja simulation. The horizontal axis represent different number of motes and the vertical axis is transmission success rates measured in ETX.

availability and granularity of spectral information improves as the number of cooperating node grows, leading to more accurate route selections.

8.4 Experiment Settings

Experiments through hardware implementation were conducted in order to provide more concrete investigation in real physical surroundings. This section provides specifications of the experimental work. The deployment of devices in the experimental environment is firstly introduced. Then different configurations for the experiments are described.

8.4.1 Deployment

Experiments were again conducted at Location A as in Chapter 7. The placement of devices was as shown in Figure 8.3. A total of nine GINA motes [13] were used and numbered accordingly at small circles in the figure. The brown circle indicates the position of Node *1* which combined with the attached desktop acted as the gateway point, whilst the remaining motes were placed at the blue circles. In order to facilitate the examination of the behaviours of RPL, an external jamming device was also utilised. Red triangles *J1* through *J3* in Figure 8.3 mark the spots where the jammer was deployed. Details about the jammer are later provided as part of the configuration.

8.4.2 Configuration

In this subsection we report a range of experimental configurations for evaluating the performance of routing metrics. Our system was implemented on top of UC Berkeley's OpenWSN [145] with A-TSCH deployed at the MAC layer as described in Chapter 7. The implementation used KFES estimator and event-triggered blacklisting. JANET IPv6 tunnelling service was installed and activated because RPL is IP-based and that 6LoWPAN is an integral part of the OpenWSN stack. Due to the limited size of the environment (approximately $12m \times 12m$),

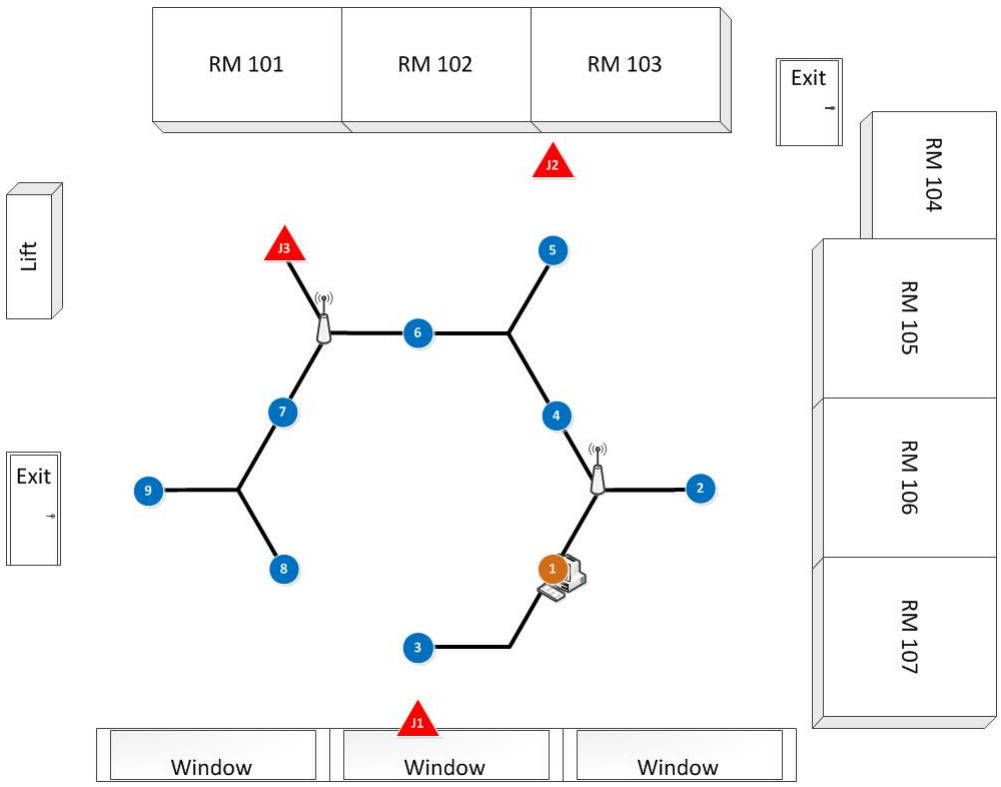


Figure 8.3: The deployment of GINA motes for the experiments of RPL. Circles denote positions of GINA motes with corresponding IDs. The gateway point was at the brown circle and the remaining motes resided at blue ones. Red triangles indicate the range of locations where wireless traffic jammer was able to be placed.

the multi-hop network structure required for evaluating routing protocols may not always be guaranteed without explicit control. To mitigate this practical issue, we explicitly configured that packets for the gateway must be forwarded by either Node *2*, *3* or *4*, forcing a multi-hop routing hierarchy.

Table 8.3: Configuration of experiments on performance of different routing metrics. Experiments were conducted using four different combinations of settings, and are hereby identified as Experiment I through IV.

Experiment ID	**I**	**II**	**III**	**IV**
Main RPL Metric	ETX	SLC	ETX	SLC
Number of Jammer Spots	N/A		3	
Jamming Duration per Spot			25 minutes	
Inter-spot Interval			5 minutes	
Packet Size	128 bytes			
Transmission Rate	1 packet per second			
Duration per Experiment	4 hours			
Iterations	5			
Total Running Time	80 hours			

Configurations used for experiments in this chapter are summarised in Table 8.3. Aiming to evaluate the performance of routing metrics in distinct scenarios, Experiment I and II, Experiment III and IV were configured in pairs. Specifically, Experiment I and II were conducted to compare ETX and SLC under normal spectral condition, whilst Experiment III and IV, on the other hand, were respectively carried out using ETX and SLC under the influence of an external jamming device in order to evaluate the performance in the scenarios of manual intervention in the spectral condition. The jammer was activated for 25 minutes at *J1*, *J2* and *J3* in a round robin fashion with a five minutes interval between

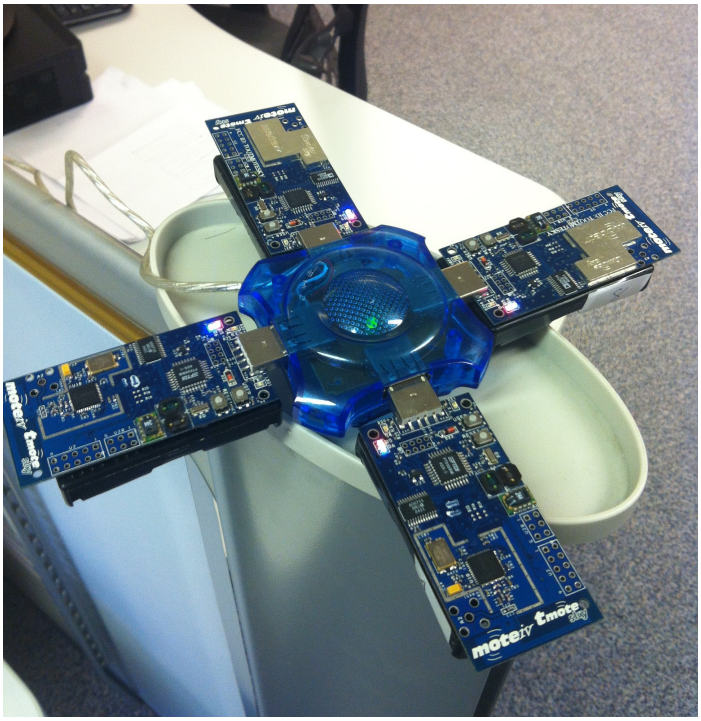


Figure 8.4: Wireless jamme built with four combined Tmote Sky motes.

successive spots.

The jammer was built with four Tmote Sky board [14], as pictured in Figure 8.4. Each Tmote board was configured with a traffic generator program using the TinyOS operating system [148]. Specifically, the jammers continuously send 100 packets at -1 dBm with inter-transmission time of 1 millisecond in four of the 16 IEEE 802.15.4 channels. Once finished, the jammer move to the next four channels and repeat the process. Note the decision of using Tmote board is not significant but purely for practical reasons such as availability and battery types.

In every session of Experiment I through IV experiments, nodes were configured to send one 128-byte packets every second to the gateway for four hours. Each experiment were repeated for five sessions, resulting in a total of 20 hours running time for each experiment.

8.5 Evaluation Criteria

The performance of default and spectral RPL metrics are evaluated in both the control plane and data plane. The control plane behaviours are examined to verify that the protocol is able to behave correctly depending on the routing metric used. The data plane is also investigated to demonstrate the effect of RPL on the network communication.

8.5.1 Control Plane Metrics

In order to examine the behaviours of the routing protocol, records of routing decisions made at individual nodes are quantified through the metric of *Path Probability* (PP), which as the name suggests is the percentage of certain neighbour being chosen as the next-hop out of all decisions made. For example PP_{A_B} is the ratio of routing decisions made by A that prefer B as the next-hop.

8.5.2 Data Plane Metric

The spectrum awareness is introduced to RPL in order to improve the performance of data transmission. Accordingly the transmission success rates are examined to investigate and compare the effect of different RPL metrics. Transmission performance is evaluated in terms of Expected Transmission Count (ETX). ETX for individual links are calculated and denoted by ETX_{P_Q} where P and Q stand for the sending and receiving ends of the link, respectively.

8.6 Experimental Results

In this section, the performance of RPL using different routing metrics are evaluated based on experimental evidence. For Experiment I and II which were carried out in normal spectral condition, we first examine the resulting PP of individual nodes achieved with both metrics to show that RPL can more efficiently choose the suitable next-hops with SLC. Then the ETX achieved at individual nodes as

well as entire network are calculated and compared to demonstrate the enhanced transmission success rates by using SLC in routing. For scenarios with jammer activated, the PP observed in Experiment IV are illustrated and compared with those observed in Experiment II to verify that SLC is able to make suitable reaction to modified spectral condition. And then ETX of Experiment III and IV are again compared to show that the adoption of SLC leads to enhanced transmission success rates.

8.6.1 Normal Spectral Condition Scenario

The results of Experiment I and II are shown in Table 8.4 and 8.5, respectively, which provide a comparison between the performance of ETX and SLC as routing metrics under normal spectral condition. Resulting transmission performances are provided for individual links of (sender,next-hop) that occurred in the experiments and the corresponding PP are shown in brackets. Performances of Node *2*, *3* and *4* were not affected by the routing process as they were directly connected to the gateway and their PP were always 100% to Node *1*. Accordingly information about these nodes are omitted in the tables and discussions hereafter focus on Node *5* to *9*.

8.6.1.1 Control Plane

It is observed in Table 8.5 that in Experiment II higher PP are uniformly associated with next-hops which deliver good transmission success rates. For instance, Node *6* has a probability of 59.40% to choose Node *4* as the next-hop which deliver the ETX of 1.013, whilst the link to Node *3* delivers a higher ETX of 1.048 and is therefore only chosen at 40.60% of the time. The similar findings can be observed for Node *7*, *8* and *9*, too.

For Experiment I where ETX was used as the routing metric, the choice of next-hops are not consistently in accord with actual link quality as observed in Table 8.4. Specifically, nodes fail to select the next-hops with lower ETX except

Table 8.4: Results of Experiment I. ETX and PP (in brackets) are averaged over all experiment iterations.

Sender \ Next-hop	2	3	4	5	6	7	8	Mean Sender ETX
5	1.015 (49.30%)		1.031 (50.70%)					1.023
6		1.041 (55.40%)	1.009 (44.60%)					1.027
7					1.018 (66.70%)		1.115 (33.30%)	1.050
8		1.01 (40.60%)	1.035 (59.40%)					1.025
9						1.002 (37.60%)	1.011 (62.40%)	1.008
Overall Network ETX	**1.027**							

Table 8.5: Results of Experiment II. ETX and PP (in brackets) are averaged over all experiment sessions.

Sender \ Next-hop	2	3	4	5	6	7	8	Mean Sender ETX
5	1.016 (56.60%)		1.022 (43.40%)					1.019
6		1.048 (40.60%)	1.013 (59.40%)					1.027
7					1.008 (66.50%)		1.113 (33.50%)	1.043
8		1.009 (60.60%)	1.02 (39.40%)					1.013
9						1.012 (12.60%)	1.011 (87.40%)	1.011
Overall Network ETX	**1.023**							

for Node *7*, which exhibits a notable contrast to the behaviour of the SLC metric in Table 8.5. This observation appears to contradict the nature of the default routing metric as it is directly related to ETX. A possible explanation for this is that the metric can only capture deterioration in link quality after packets have already been lost, and this delay means that routing decisions may lag behind the actually situation and leads to suboptimal next-hop selection.

8.6.1.2 Data Plane

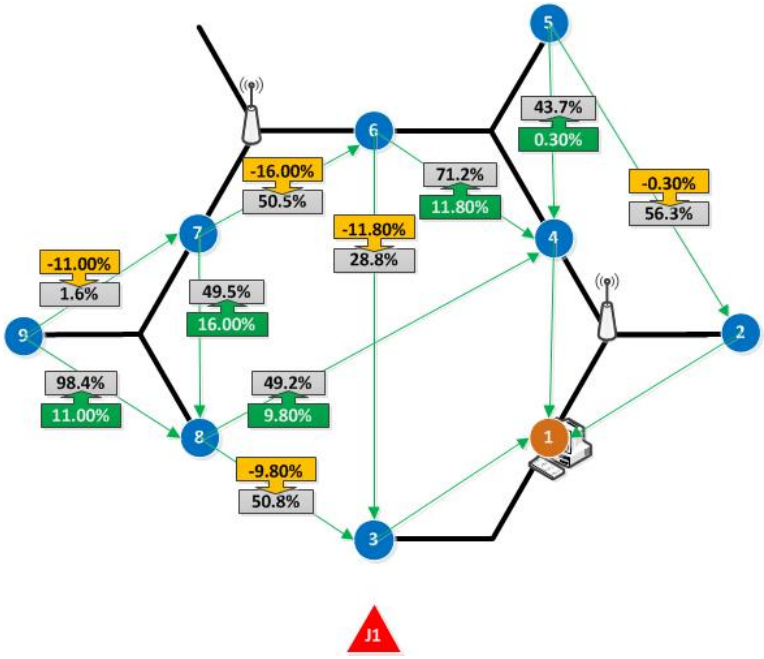
Data in Table 8.4 and Table 8.5 also provide information for illustrating the effect of the spectral routing metric on transmission performance. The mean ETX of each sender is provided in the rightmost columns of every row, which is calculated as the weighted mean of ETX with respect to corresponding PP.

It can be observed that the majority of individual node ETX are lower in Table 8.5, with the only exception of Node *6* where the ETX are same in both tables. This shows that the transmission performance is enhanced at almost all nodes with routing using SLC. Furthermore, the overall ETX of the entire network displayed in the last rows of Table 8.4 and Table 8.5 also lead to the same observation.

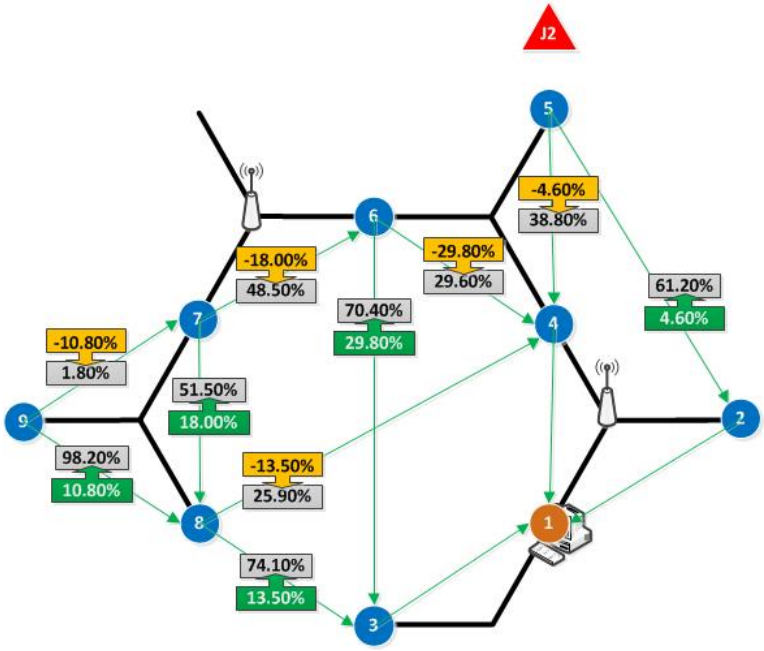
8.6.2 Jamming Scenarios

8.6.2.1 Control Plane

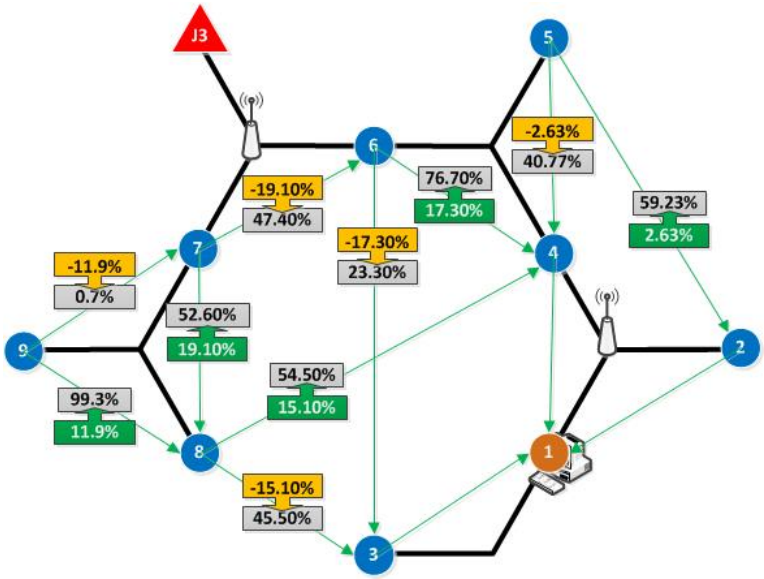
Next we examine the behaviour of RPL using SLC when the jammer was deployed in the environment. Results of Experiment IV are illustrated and compared with Experiment II in Figure 8.5a, 8.5b and 8.5c for spot *J1*, *J2* and *J3*, respectively. In these figures, the PP of links established in the experiment IV are specified in grey boxes. Green boxes with upward arrows denote the percentage of growth in the corresponding PP observed in Experiment II where no jammer was utilised, whilst yellow boxes with downward arrows indicate the percentage of reduction.


5
6
43.7%
0.30%
-16.00%
50.5%
71.2%
11.80%
-0.30%
56.3%
7
-11.80%
28.8%
4
-11.00%
1.6%
49.5%
16.00%
9
2
98.4%
11.00%
49.2%
9.80%
8
1
-9.80%
50.8%
3
J1

(a)


J2
5
6
-4.60%
38.80%
-18.00%
48.50%
-29.80%
29.60%
61.20%
4.60%
7
70.40%
29.80%
4
-10.80%
1.80%
51.50%
18.00%
9
2
98.20%
10.80%
-13.50%
25.90%
8
1
74.10%
13.50%
3

(b)



(c)

Figure 8.5: Changes in PP under the influence jammer placed at spot *J1*, *J2* and *J3*, respectively. Link PP are provided in grey boxes, and changes relative to the scenario without jammer are specified in green and yellow boxes, respectively.

The jammer placed at *J1* is expected to affect Node *3* most significantly due to the close proximity. In Figure 8.5a, RPL is found to have correctly reacted to this situation. Specifically, the PP for links to Node *3* are reduced by 11.8% and 9.8% at Node *6* and *8*, respectively. The increased PP to Node *8* from Node *7* and *9* may not be accurately explained in this figure. It is possible that this is a reduction in the accumulative link cost of the entire path from Node *8* to the gateway. The fact that both Node *7* and *9* are found to agree with this change to some extent shows that this change is not random fluctuation at isolated nodes.

When the jammer is activated at *J2*, Node *5* should be the most affected. In Figure 8.5b, however, Node *5* has never been chosen as the next-hop by any nodes, and the effect of jamming signals cannot be reflected by looking at this node. However, nearby Node *6* shows a PP decreased by 18% from to Node *7*, which correspond to the anticipated effect of the jammer. Similarly, Node *6* and

8 are also observed to tend to avoid Node *4* with PP reduction of 29.8% and 13.5%, respectively.

Finally, as shown in Figure 8.5c, Node *6* and *7* are most proximate to jammer at at *J3*. In accord with the scenario, the PP to Node *6* has decreased by 19.1% and that PP to Node *7* has reduced by 11.9%. This observation is consistent with previous findings for *J1* and *J2*, and these show that RPL using SLC metric is able to bahave according to spectral condition as expected.

8.6.2.2 Data Plane

Table 8.6: A comparison of transmission performance achieved using ETX-based and SLC-based routing metrics in jamming scenarios where jammer was placed at J1, J2 and J3, respectively.

Jammer Spot / **RPL metric**	**J1**	**J2**	**J3**
ETX-based	1.023	1.024	1.018
SLC-based	1.020	1.018	1.013

The overall network ETX of Experiment III and IV are shown in Table 8.6 with respective to jammer spots, providing a comparison of data plane performance for the ETX and SLC routing metrics in the jamming scenario. It can be observed that SLC-based RPL provides lower ETX than default RPL in all three jammer scenarios, which confirms that RPL employing SLC is able to improved transmission success rates.

8.7 Summary

This chapter provides an exploration of incorporating spectral knowledge of MAC layer into routing process at the network layer. First, the RPL routing protocol

for low-power wireless networks was reviewed. Then *Spectral Link Cost* (SLC) was introduced as a new routing metric that can be employed by RPL to take into account potential communication costs associated with ambient noise when determining the next-hops. The factors of Disparity (DI) and Exhaustion (EH) are formally defined along with their composition into SLC. Then we described and conducted experiments to evaluate the effectiveness of the proposed cross-layer synergy between blacklisting and routing in two different scenarios. Experimental results confirm that RPL employing SLC behaves as expected and provide an enhanced reliability in network communication. Although the extent of improvement is not as significant as those delivered by A-TSCH, the work on SLC is still at an early stage and points out an interesting direction.

Chapter 9

Conclusions and Future Work

In this thesis, we have presented the Adaptive Time-slotted Channel Hopping (A-TSCH) framework which protects the transmission performance of low-power wireless networks from the influence of ambient noise which is one of the main challenges as a result of the proliferation of wireless communication in the vision of IoT. The thesis also explored the synergy between A-TSCH and RPL routing protocols which further mitigates risks and overheads caused by interference. In this chapter, we first summarise key aspects of the thesis in Section 9.1. Then contributions of the thesis are revisited in Section 9.2. Limitations of work presented in this thesis are identified in Section 9.3. Finally, we conclude this chapter and the thesis with a dicussion of directions for further researches in Section 9.4.

9.1 Thesis Summary

Target Area and Motivation In Chapter 1, the notion of IoT was introduced along with its characteristics and associated challenges. Taking into account of the increasingly acute contest for limited spectrum resource, especially in the license-free Industrial, Scientific and Medical (ISM) band where many IoT devices operate, it was argued that the ability to make dynamic adaptation to communication behaviours was vital for the prospect of IoT. Accordingly, a variety of techniques for adaptive communication were surveyed in Chapter 2, revealing

adaptation opportunities at the PHY, MAC and the network layer of the network model. As a consequence, this thesis has been motivated by the need to address challenges facing IoT and the opportunities for adaptive solutions that can be devised at these layers.

Design Rationale The framework of A-TSCH as described in Chapter 3 was designed based on the rationale that the network performance can be protected by avoiding the parts of spectral band with high ambient noise. In Chapter 4, this rationale was investigated through statistical analysis of experimental results. Specifically, communication traces were collected at different locations whilst the transmitted packets and noise levels were being monitored. Then the relationship between communication performance and corresponding ambient noise were examined, demonstrating that the achieved transmission success rates were negatively related to the ambient noise levels. It was also noticed that the influence of noise on transmission was only notable when it is above certain level. Accordingly, through statistical analysis, potential thresholds of ambient noise were suggested, above which transmission was found to display significant deterioration.

Channel Desirability Key to the proposed framework is the ability to formulate knowledge about channel desirability based on their noise profiles. Recognising that periodically sampled ambient noise may not be sufficient to reflect the spectral condition, estimation techniques were discussed in Chapter 5 to mitigate the limitation. Specifically, Exponential Smoothing (ES), Kalman filter (KF) and an additional KF-aided ES (KFES) scheme have been investigated. By carrying simulations in Matlab, the operating parameters of the estimators were tuned and a preliminary comparison of their performance was presented.

Blacklisting Using desirability information formulated through estimators, the subset of undesirable channels can be identified and their IDs are stored in a data structure called blacklist. Then the network abandons the use of channels included in the blacklist. The generation of blacklists and the corresponding adaptation

to communication behaviours are collectively referred to as blacklisting.

Blacklisting operation in A-TSCH was detailed in Chapter 6. The election of channels to blacklist in our framework is carried out in a four-phase cooperation. The cooperative blacklisting is organised in a cluster-based network architecture as reviewed in Chapter 2. Participating node first independently come up with their blacklist candidates in the Candidate Identification stage. During the subsequent Candidate Aggregation phase, these candidates converge at corresponding clusterheads, where the blacklist to be formally used for the cluster is yielded in the Blacklist Decision-making phase. Finally the decision is broadcast to cluster members in the Synchronisation stage.

When communicating, nodes first checks the operating frequency against the blacklist and replaces it with an pseudo-randomly generated alternative if it is found blacklisted, therefore avoiding negative effect of undesirable frequencies. However, channels exhibiting undesirable features at one end of the link may appear harmless at the other, due to the spatial diversity of spectral condition. As a result, the cluster-wide blacklist yielded by the clusterhead cannot cater every particular condition experienced by each node. To alleviate this problem, cluster members postpone scheduled transmission if the operating channel is legitimate according to the cluster blacklist but locally undesirable, as long as the outgoing buffer is not getting overflown.

Spectral Awareness in Routing The synergy between A-TSCH and RPL was explored in Chapter 8 in order to make routing process aware of the spectral condition. Each node's blacklist is now considered a manifestation of its spectral characteristics, and the metric of Spectral Link Cost (SLC) is calculated based on the blacklists at both ends of a link. By incorporating the Disparity (DI) and Exhaustion (EH) factors, the SLC takes into account both the similarity and combined size of the blacklists at neighbouring nodes. As a result, the more similar and smaller the blacklists are, the more preference is given to the route.

Compared with default ETX-based metric of RPL, SLC was found to be more efficient in preventing performance deterioration caused by ambient noise. Changing spectral conditions can be effectively reflected through SLC allowing RPL to update routing decisions accordingly, whereas the ETX-based metric is only capable of reflecting this fact after packets had already been lost.

9.2 Summary of Contributions

The following contributions are made in this thesis:

(i) The Adaptive Time-slotted Channel Hopping (A-TSCH) framework was developed to facilitate adaptive communication at the MAC layer. Extending the existing TSCH of IEEE 802.15.4e standard, A-TSCH enables low-power wireless networks greater control over their channel hopping behaviour. Through learning the ambient noise floors of specific channels (Chapter 5) and blacklisting technique (Chapter 6), A-TSCH restricts channel hopping sequence to the subset of channels exhibiting desirable noise characteristics. As a result, risks of potential packet loss caused by interferences are reduced, whilst TSCH's benefit of improved robustness remains preserved.

(ii) An algorithm for Cooperative Blacklisting (CB) is developed in Chapter 6. Because of the inherent spatial diversity of spectral condition, blacklisting decisions independently made by individual nodes are not expected to be accurate for other parts of the network. As communications inevitably involve multiple entities, a suitable solution is to make blacklisting decisions through cooperation. The CB module employs cluster-wise cooperation strategy, which is carried out in phases detailed in Section 6.3. Compared with the global and link-wise strategies discussed in Section 6.2, the cluster-wise cooperation represents a balance between blacklisting accuracy and overhead incurred by blacklisting.

(iii) An investigation of estimation techniques was conducted in Chapter 5 for the Channel Quality Estimation (CQE) module which is responsible for learning spectral condition of individual channels. Specifically, Exponential Smoothing (ES) and Kalman filter (KF) were considered as possible estimators for transforming ambient noise readings containing momentary fluctuation into stable channel quality estimates. Different variants of ES and KF were adjusted for our application and tuned through simulation using communication traces introduced in Chapter 4. Estimates produced by CQE provide the core information that the functions of the CB module depends upon.

(iv) Evidences for the performance of A-TSCH was provided through experiments in Chapter 7. The framework was implemented atop UC Berkerley's OpenWSN protocol stack and installed in Guidance and Inertial Navigation Assistant (GINA) motes. Experiments were carried out in a typical workplace environment with a notable presence of wireless interference. Different configurations including blacklist size, invocation mechanism and the estimator were considered and the performance of A-TSCH was assessed in both the control and data planes. Results show that A-TSCH operated correctly and a notable improvement in transmission success rates is observed.

(v) A routing metric Spectral Link Cost (SLC) was devised in Chapter 8 to exploit the synergy between A-TSCH and the RPL routing protocol. RPL provides the architecture of routing functionality without imposing rigid restriction on specific metric. Taking advantage of the flexibility of RPL, SLC is to incorporate spectral awareness in routing. As defined in Section 8.2, SLC is calculated on the diversity of spectral condition and the overall desirable channels considering both ends of any potential route. Accordingly, risks and overhead for networks using A-TSCH in the presence of ambient noise are taken into account in routing.

(vi) The performance of SLC was evaluated in both simulation (Section 8.3)

and experiments (Section 8.6). The Cooja simulator of Contiki was used to model networks using RPL, which allowed testing with a range of different mote numbers and density. RPL was also implemented in GINA motes and experimented in typical workplace environment to strengthen our findings. An external jamming device was also used to inspect the behaviour of RPL under drastic deterioration of certain channels. Evaluation of simulation and experimental data were conducted by comparing the performance of RPL using both SLC and the default ETX-based metrics. It was observed that transmission success rates were improved when SLC was used by RPL, indicating the effectiveness of this cross-layer enhancement approach.

9.3 Limitations

A number of limitations of this work are identified:

- Due to the constraints of the hardware platform, the spectral condition is observed periodically during a fixed time-window and therefore the spectrum is not continuously monitored. As a consequence, very high frequency changes in the wireless medium may remain undetected.

- We only consider the paradigm of Multipoint-to-Point (MP2P) upstream data transmission, which represents typical cases of low-power wireless networks. In this scenario data generated at individual nodes travel towards the gateway (sink). Although alternative paradigms of Point-To-Multipoint (P2MP) and Point-To-Point (P2P) would not affect the effectiveness of the proposed framework, they may create problems for the routing mechanism of Chapter 8 which is beyond the scope of this work.

- The decision-making entities of the cooperative blacklisting algorithm described in Chapter 6 are the clusterheads to which cluster members send data. In other words, the clustered hierarchy of blacklisting in the control plane is identical with that of data transmission in the data plane, which

may not be the optimal arrangement in reality. Alternatively, more sophisticated mechanisms could be adopted such that clusters are composed differently for data communication and blacklisting cooperation, and nodes that receives data from cluster members are not necessarily those responsible for making blacklisting decisions.

- Discussion on the impact of A-TSCH and SLC on IoT networks are mainly based on ETX metric. It would be useful to take into account other metrics in terms of network performance such as throughput or delay.

9.4 Directions for Future Work

Noise Floor Listening One of the most acute challenges for low-power networks is the restricted life-time of these devices that are typically powered by batteries. It is known that the most costly operation in terms of energy consumption is radio communication, and the life-time of the networks can be notably prolonged by minimising radio operations [149, 150, 151]. In A-TSCH, the most frequently conducted radio operation is the Noise Floor Listening which carries out spectrum sensing in two NF timeslots of every slotframe. Although no data are sent in the NF slots, switching on the radio is still a relatively expensive operation compared with computation instructions in the microprocessor. In order to reduce the number of spectrum sensing operation, the following directions can be investigated:

(i) Balancing the workload of spectrum sensing among all cluster members. Instead of having to sense each of the 16 IEEE 802.15.4 channels, the clusterhead can allocate certain parts of the 2.4 GHz band for individual members to detect. As a result nodes in some of the NF slots can remain idle, reducing energy consumption. This approach needs to take into account the behaviour of individual cluster members. For example, if the candidate blacklists reported by certain node are notably changeable, then it is not suitable to reduce its range of detection channels.

(ii) Reducing the frequency of noise floor listening for stable nodes. If the candidate blacklists from certain nodes are found to be largely stable, then the rate of spectrum sensing can be reduced besides the number of channels to detect. For instance, the rate of spectrum sensing can be halved by skipping every other 16 NF slots.

Channel Desirability Estimation In order to ensure the effectiveness of any spectrum-related adaptation, the channel desirability knowledge yielded through Channel Quality Estimation should be able to accurately reflect the spectral condition. We propose the following aspects which can be explored to improve the quality of channel desirability estimates on the basis of this work:

(iii) Further investigation into estimators. Although experiments have provided evidence that A-TSCH can correctly identify noisy channels and enhance transmission performance, no deterministic findings have been drawn about the effectiveness of three estimators discussed in this work. Accordingly, existing estimators particularly the KFES should be examined for further adjustment. Moreover, more estimation methods should be explored to help improve or replace currently used methods.

(iv) Incorporating additional type of information in representing desirability of channels. For example, packet delivery ratio (PDR) records can be kept for individual channels. Although PDR only captures changes in channel quality after deterioration has already taken place, it can provide important complementary indication of channel desirability, especially for non-spectral factors such as packet loss caused by physical obstacles.

Blacklisting Algorithm Mechanisms for identifying channels to blacklist are another focal point for future work. Currently in A-TSCH, channels are solely selected based on noise thresholds derived from statistical studies of the relationship between noise strength and transmission success rates. In the following, we suggests a number of complementary blacklisting criteria for future investigations:

(v) Flexible threshold relative to overall noise levels. Currently, the noise thresholds are used in A-TSCH without considering the noise profile specific to the particular environment. This leaves a potentially problematic scenario where the blacklisting criteria become ineffective when the overall ambient noise level of the 2.4 GHz band are uniformly higher or lower than the thresholds. A possible approach to address this issues is to adapt the noise threshold with respect to the overall noise strength. Accordingly, a direction for future work is to design a suitable scheme which dynamically adjusts the threshold to maintain the benefit of A-TSCH.

(vi) Integrating channel reputation into decision-making. Using threshold-based blacklisting criteria, the decisions may still be unstable if the noise levels of certain channels tend to fluctuate around the threshold. Although this may not necessarily affect the accuracy of the decisions, rapidly changing blacklists can incur a series of overhead and increase the risk of desynchronisation. Accordingly, a balance between blacklisting accuracy and system stability should be considered. A possible solution is to introduce an additional channel reputation metric, which reflects the history of blacklisting about certain frequencies. This new metric is anticipated to create an inertial factor in decision-making, thus improving the stability of blacklist.

Additional Opportunities for Adaptive Communication Apart from aspects already involved in this work, some additional opportunities of adaptations in communication behaviours for low-power wireless networks are identified:

(vii) Adaptive transmission power. As discussed earlier, the power budget is an important issue for low-power networks and the main consumer of device energy is radio transmission. Because the strength of transmission power is related to the ability of the signal to resist the effect of ambient noise [152], we anticipate that the knowledge of spectral condition can be utilised to adapt the power consumed by the radio. Specifically, after the antenna is tuned in for the selected operating channels, the transmission

power is dynamically adjusted according to the corresponding estimated ambient noise level. The transmission power is decrease to reduce power consumption when the channel is observed to be quiet, and increased to maintain delivery rate when the noise level is high.

(viii) Responsive data rates. The rate at which outgoing traffics are generated can also be related to the underlying factors in PHY and MAC layers. By devising an indicator that takes into account the information such as the size of blacklist, transmission power, length of outgoing queue, and so forth, the data rate can be adapted accordingly to best suit the capability of the network to communicate.

Bibliography

[1] V. Ovidiu, M. Harrison, H. Vogt, K. Kalaboukas, M. Tomasella, K. Wouters, S. Gusmeroli, and S. Haller, *Internet of Things Strategic Research Roadmap.* European Commission - Information Society and Media DG, 2009.

[2] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the Internet of Things* (D. Uckelmann, M. Harrison, and F. Michahelles, eds.), pp. 1–24, Springer Berlin Heidelberg, 2011.

[3] S. Siorpaes, G. Broll, M. Paolucci, E. Rukzio, J. Hamard, M. Wagner, A. Schmidt, and D. Eurolabs, "Mobile interaction with the internet of things," in *In Adjunct Proc. of Pervasive 2006 Late Breaking Results*, 2006.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, pp. 2787–2805, October 2010.

[5] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pp. 1 –6, sept. 2011.

[6] M. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pp. 21 –26, may 2012.

[7] ITU, "Itu internet reports 2005: The internet of things," tech. rep., ITU, 2005.

[8] H. Sundmaeker, P. Guillemin, and P. Friess, eds., *Vision and Challenges for realising the Internet of Things*. Publications Office of the European Union, 2010.

[9] L. Tan and N. Wang, "Future internet: The internet of things," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 5, pp. V5–376–V5–380, 2010.

[10] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *Communications Magazine, IEEE*, vol. 49, pp. 68 –75, november 2011.

[11] X. M. Zhang and N. Zhang, "An open, secure and flexible platform based on internet of things and cloud computing for ambient aiding living and telemedicine," in *Computer and Management (CAMAN), 2011 International Conference on*, pp. 1 –4, may 2011.

[12] P. Du and G. Roussos, "Adaptive channel hopping for wireless sensor networks," in *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, 2011.

[13] WARPwING the Wireless Autonomous Robot Platform with Inertial Navigation and Guidance, WARPwING. [online]. Accessed: 20 November 2011.

[14] TmoteSky IEEE 802.15.4 Development Kit, Texas Instruments. [online]. Accessed: 15 March 2010.

[15] R. Musaloiu-E. and A. Terzis, "Minimising the effect of wifi interference in 802.15.4 wireless sensor networks," *Int. J. Sen. Netw.*, vol. 3, pp. 43–54, Dec. 2008.

[16] J. D. Day and H. Zimmermann, "The osi reference model," in *Conformance testing methodologies and architectures for OSI protocols* (R. J. Linn and

M. U. Uyar, eds.), ch. The OSI reference model, pp. 38–44, Los Alamitos, CA, USA: IEEE Computer Society Press, 1995.

[17] B. M. Leiner, R. Cole, J. Postel, and D. Mills, "Computer communications (3rd ed.)," in *Computer communications (3rd ed.)* (W. Stallings, ed.), ch. The DARPA internet protocol suite, pp. 48–53, Piscataway, NJ, USA: IEEE Press, 1992.

[18] K.-C. Chen, Y.-J. Peng, N. Prasad, Y.-C. Liang, and S. Sun, "Cognitive radio network architecture: part ii – trusted network layer structure," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, ICUIMC '08, (New York, NY, USA), pp. 120–124, ACM, 2008.

[19] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley Publishing, 2010.

[20] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1, pp. 772 – 776 Vol.1, nov. 2004.

[21] D. Cabric and R. W. Brodersen, "Physical layer design issues unique to cognitive radio systems," in *16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, (PIMRC 2005)*, September 2005.

[22] Z. Tian and G. B. Giannakis, "Compressed sensing for wideband cognitive radios," *2007 IEEE International Conference on Acoustics Speech and Signal Processing ICASSP 07*, vol. 4, pp. IV–1357–IV–1360, 2007.

[23] J. Lundén, V. Koivunen, A. Huttunen, and H. V. Poor, "Collaborative cyclostationary spectrum sensing for cognitive radio systems," *Trans. Sig. Proc.*, vol. 57, pp. 4182–4195, November 2009.

[24] K. Ben Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proceedings of the IEEE*, vol. 97, pp. 878 – 893, 2009.

[25] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrum sensing for cognitive radio: challenges and solutions," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 2:2–2:2, jan 2010.

[26] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Comm.*, vol. 4, no. 1, pp. 40 – 62, 2011.

[27] D. Cabric, A. Tkachenko, and R. Brodersen, "Spectrum sensing measurements of pilot, energy, and collaborative detection," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1 –7, oct. 2006.

[28] I. Budiarjo, M. K. Lakshmanan, and H. Nikookar, "Cognitive radio dynamic access techniques," *Wirel. Pers. Commun.*, vol. 45, pp. 293–324, May 2008.

[29] A. Ghasemi and E. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Commun. Mag.*, vol. 46, pp. 32–39, 2008.

[30] A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, 2005.

[31] F. F. Digham, M. S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *Communications, IEEE Transactions on*, vol. 55, no. 1, pp. 21 – 24, 2007.

[32] H. Kim and K. G. Shin, "In-band spectrum sensing in cognitive radio networks: energy detection or feature detection?," in *Proc. 14th ACM international conference on Mobile computing and networking*, MobiCom '08, (New York, NY, USA), pp. 14–25, ACM, 2008.

[33] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, vol. 11, pp. 116 – 130, 2009.

[34] Y. Zeng and Y.-C. Liang, "Maximum-minimum eigenvalue detection for cognitive radio," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pp. 1 –5, sept. 2007.

[35] Y. Zeng and Y. chang Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," *Communications, IEEE Transactions on*, vol. 57, pp. 1784 –1793, june 2009.

[36] F. Penna, R. Garello, and M. Spirito, "Cooperative spectrum sensing based on the limiting eigenvalue ratio distribution in wishart matrices," *Communications Letters, IEEE*, vol. 13, pp. 507 –509, july 2009.

[37] A. Kortun, T. Ratnarajah, M. Sellathurai, and C. Zhong, "On the performance of eigenvalue-based spectrum sensing for cognitive radio," in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pp. 1 –6, april 2010.

[38] O. B. Akan, O. B. Karli, and O. Ergul, "Cognitive radio sensor networks," *Netwrk. Mag. of Global Internetwkg.*, vol. 23, pp. 34–40, July 2009.

[39] D. Cabric, A. Tkachenko, and R. W. Brodersen, "Experimental study of spectrum sensing based on energy detection and network cooperation," in *Proceedings of the first international workshop on Technology and policy for accessing spectrum*, TAPAS '06, (New York, NY, USA), ACM, 2006.

[40] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Optimal multiband joint detection for spectrum sensing in cognitive radio networks," *Trans. Sig. Proc.*, vol. 57, pp. 1128–1140, March 2009.

[41] K. Sithamparanathan, *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation, and Localization*. Artech House, 2012.

[42] L. Lu, H.-C. Wu, and S. Iyengar, "A novel robust detection algorithm for spectrum sensing," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 2, pp. 305–315, 2011.

[43] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 201–220, 2005.

[44] F. K. Jondral, "Software-defined radio: basics and evolution to cognitive radio," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 3, pp. 275–283, 2005.

[45] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: why channel hopping makes sense," in *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, PE-WASUN '09, (New York, NY, USA), pp. 116–123, ACM, 2009.

[46] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-part i: system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[47] G. Ganesan and Y. Li, "Agility improvement through cooperative diversity in cognitive radio," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 5, pp. 5 pp. –2509, dec. 2005.

[48] S. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 4, pp. 1658 –1663, june 2006.

[49] Z. Quan, S. Cui, and A. H. Sayed, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 28 – 40, 2008.

[50] V. Sharma and A. Jayaprakasam, "An efficient algorithm for cooperative spectrum sensing in cognitive radio networks," *Change*, vol. 5, no. 1, p. 6, 2008.

[51] E. C. Y. Peh, Y.-C. Liang, and Y. L. Guan, "Optimization of cooperative sensing in cognitive radio networks: a sensing-throughput tradeoff view," in *Proceedings of the 2009 IEEE international conference on Communications*, ICC'09, (Piscataway, NJ, USA), pp. 3521–3525, IEEE Press, 2009.

[52] F. Gao, W. Yuan, W. Liu, W. Cheng, and S. Wang, "Pipelined cooperative spectrum sensing in cognitive radio networks," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pp. 1 –5, april 2009.

[53] R. Fan and H. Jiang, "Optimal multi-channel cooperative sensing in cognitive radio networks," *Trans. Wireless. Comm.*, vol. 9, pp. 1128–1138, March 2010.

[54] W. Yue, B. Zheng, J. Cui, S. Tang, and L. Wang, "Improved cooperative spectrum sensing for cognitive radio under bandwidth constraints," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, (New York, NY, USA), pp. 843–847, ACM, 2010.

[55] Y. Zhao, M. Song, and C. Xin, "A weighted cooperative spectrum sensing framework for infrastructure-based cognitive radio networks," *Comput. Commun.*, vol. 34, pp. 1510–1517, August 2011.

[56] H. Khaleel, F. Penna, C. Pastrone, R. Tomasi, and M. Spirito, "Distributed spectrum sensing and channel selection in opportunistic wireless personal area networks," in *Proc. 2nd Int. Workshop on Mobile Opportunistic Networking*, MobiOpp '10, (New York, NY, USA), pp. 185–187, ACM, 2010.

[57] T. M. N. Ngatched, A. S. Alfa, and J. Cai, "Cooperative sensing with transmit diversity based on randomized stbc in cr networks," in *Proceedings*

of the 6th International Wireless Communications and Mobile Computing Conference, IWCMC '10, (New York, NY, USA), pp. 225–230, ACM, 2010.

[58] N. Vlajic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, pp. 9 pp. –268, 0-0 2006.

[59] I. Sim, K. Choi, K. Kwon, and J. Lee, "Energy efficient cluster header selection algorithm in wsn," in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on*, pp. 584 –587, march 2009.

[60] C. Diallo, M. Marot, and M. Becker, "Single-node cluster reduction in wsn and energy-efficiency during cluster formation," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, pp. 1 –10, june 2010.

[61] L. Guo, F. Chen, Z. Dai, and Z. Liu, "Wsn cluster head selection algorithm based on neural network," in *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, pp. 258 –260, april 2010.

[62] H. Abusaimeh and S.-H. Yang, "Energy-aware optimization of the number of clusters and cluster-heads in wsn," in *Innovations in Information Technology (IIT), 2012 International Conference on*, pp. 178 –183, march 2012.

[63] S. M. Hosseinirad and S. Basu, "Imperialist approach to cluster head selection in wsn," *IJCA Special Issue on Wireless Communication and Mobile Networks*, vol. wcmn, pp. 1–5, January 2012. Published by Foundation of Computer Science, New York, USA.

[64] M. Nasim, S. Qaisar, and S. Lee, "An energy efficient cooperative hierarchical mimo clustering scheme for wireless sensor networks," *Sensors*, vol. 12, pp. 92–114, 2012.

[65] M. Anisi, A. Abdullah, S. Razak, and M. Ngadi, "Overview of data routing approaches for wireless sensor networks," *Sensors*, vol. 12, pp. 3964–3996, 2012.

[66] D. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *Communications, IEEE Transactions on*, vol. 29, pp. 1694 – 1701, nov 1981.

[67] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, pp. 193–204, April 2002.

[68] M. Gerla and J. T. chieh Tsai, "Multicluster, mobile, multimedia radio network," *Journal of Wireless Networks*, vol. 1, pp. 255–265, 1995.

[69] A. Parekh, "Selecting routers in ad hoc networks," in *Proceedings SBT/IEEE Intl Telecommunications Symposium*, 1994.

[70] C.-F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, "Energy efficient design of wireless ad hoc networks," in *Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*, NETWORKING '02, (London, UK, UK), pp. 376–386, Springer-Verlag, 2002.

[71] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "Topology management in cogmesh: A cluster-based cognitive radio mesh network," in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 6516 –6521, june 2007.

[72] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, "Mac-layer scheduling in cognitive radio based multi-hop wireless networks," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, WOWMOM '06, (Washington, DC, USA), pp. 191–202, IEEE Computer Society, 2006.

[73] C. Cormio and K. R. Chowdhury, "A survey on mac protocols for cognitive radio networks," *Ad Hoc Netw.*, vol. 7, pp. 1315–1329, September 2009.

[74] K. S. J. Pister and L. Doherty, "Tsmp: Time synchronized mesh protocol," in *In Proceedings of the IASTED International Symposium on Distributed Sensor Networks (DSN08*, 2008.

[75] H. Khaleel, C. Pastrone, F. Penna, M. Spirito, and R. Garello, "Impact of wi-fi traffic on the ieee 802.15.4 channels occupation in indoor environments," in *Electromagnetics in Advanced Applications, 2009. ICEAA '09. International Conference on*, pp. 1042 –1045, sept. 2009.

[76] P. Du and G. Roussos, "Adaptive time slotted channel hopping for wireless sensor networks," in *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, pp. 29 –34, sept. 2012.

[77] S. BIN and K. KYUNG SUP, "Soft combination schemes for cooperative spectrum sensing in cognitive radio networks," *ETRI journal*, vol. 31, pp. 263–270, 2009.

[78] C. da Silva, B. Choi, and K. Kim, "Distributed spectrum sensing for cognitive radio systems," in *Information Theory and Applications Workshop, 2007*, pp. 120 –123, 29 2007-feb. 2 2007.

[79] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks 2005 DySPAN 2005*, pp. 131–136, 2005.

[80] X. Zhou, J. Ma, G. Li, Y. H. Kwon, and A. Soong, "Probability-based combination for cooperative spectrum sensing in cognitive radio networks," in *Communications, 2009. ICC '09. IEEE International Conference on*, pp. 1 –5, june 2009.

[81] N. Pratas, N. Marchetti, A. Rodrigues, and R. Prasad, "Capacity limits introduced by data fusion on cooperative spectrum sensing under correlated environments," in *Communications (COMM), 2010 8th International Conference on*, 2010.

[82] J. Ma, G. Zhao, and Y. Li, "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4502 – 4507, 2008.

[83] S. Kyperountas, N. Correal, and Q. Shi, "A comparison of fusion rules for cooperative spectrum sensing in fading channels," in *Wireless Symposium and Summer School*, 2010.

[84] C. Sun, W. Zhang, and K. Ben, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 2511 –2515, june 2007.

[85] E. Visotsky, S. Kuffner, and R. Peterson, "On collaborative detection of tv transmissions in support of dynamic spectrum sharing," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 338 –345, nov. 2005.

[86] W. Zhang, R. Mallik, and K. Ben Letaief, "Cooperative spectrum sensing optimization in cognitive radio networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 3411 –3415, may 2008.

[87] A. Min, X. Zhang, and K. Shin, "Spatio-temporal fusion for small-scale primary detection in cognitive radio networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1 –5, march 2010.

[88] Z. Quan, S. Cui, and A. Sayed, "An optimal strategy for cooperative spectrum sensing in cognitive radio networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 2947 –2951, nov. 2007.

[89] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 1876 –1884, april 2008.

[90] J. Zhu, Z. Xu, F. Wang, B. Huang, and B. Zhang, "Double threshold energy detection of cooperative spectrum sensing in cognitive radio," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, pp. 1 –5, may 2008.

[91] Y. Alemseged, C. Sun, H. N. Tran, and H. Harada, "Distributed spectrum sensing with two-stage detection for cognitive radio," in *70th Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE*, 2009.

[92] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1713 – 1723 vol.3, march-3 april 2003.

[93] X. CHEN, Z. song BIE, and W. ling WU, "Detection efficiency of cooperative spectrum sensing in cognitive radio network," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 3, pp. 1 – 7, 2008.

[94] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Physical Comm.*, vol. 4, no. 1, pp. 26 – 39, 2011.

[95] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 4024 –4036, august 2009.

[96] K. Butterworth, K. Sowerby, and A. Williamson, "Correlated shadowing in an in-building propagation environment," *Electronics Letters*, vol. 33, pp. 420 –422, feb 1997.

[97] J. Shen, S. Liu, R. Zhang, and Y. Liu, "Soft versus hard cooperative energy detection under low snr," in *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pp. 128 –131, aug. 2008.

[98] R. Bruno, M. Conti, and A. Passarella, "Opportunistic networking overlays for ICT services in crisis management," in *International Conference on Information Systems for Crisis Response and Management (ISCRAM 2008). Washington, DC, USA*, 2008.

[99] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, (New York, NY, USA), pp. 133–144, ACM, 2005.

[100] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, pp. 134 –141, november 2006.

[101] L. Lilien, Z. H. Kamal, V. Bhuse, A. Gupta, and W. (wireless Sensornet Lab, "Opportunistic networks: The concept and research," in *Challenges in Privacy and Security, Proc. NSF Intl. Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN 2006)*, 2006.

[102] M. Cesana, F. Cuomo, and E. Ekici, "Routing in cognitive radio networks: Challenges and solutions," *Ad Hoc Netw.*, vol. 9, pp. 228–248, May 2011.

[103] A. Talay and D. Altilar, "Ropcorn: Routing protocol for cognitive radio ad hoc networks," in *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pp. 1 –6, oct. 2009.

[104] *SAMER: Spectrum Aware Mesh Routing in Cognitive Radio Networks*, 2008.

[105] L. Ding, T. Melodia, S. Batalama, J. Matyjas, and M. Medley, "Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 1969 –1979, may 2010.

[106] H. Ma, L. Zheng, X. Ma, and Y. luo, "Spectrum aware routing for multi-hop cognitive radio networks with a single transceiver," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, pp. 1 –6, may 2008.

[107] A. S. Cacciapuoti, M. Caleffi, and L. Paura, "Reactive routing for mobile cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 0, pp. –, 2011.

[108] A. Sampath, L. Yang, L. Cao, H. Zheng, and B. Zhao, "Invited paper: High throughput spectrum-aware routing for cognitive radio networks," *3rd International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications*, May 2008.

[109] C.-F. Shih, W. Liao, and H.-L. Chao, "Joint routing and spectrum allocation for multi-hop cognitive radio networks with route robustness consideration," *Wireless Communications, IEEE Transactions on*, vol. 10, pp. 2940 –2949, september 2011.

[110] C. Zheng, R. P. Liu, X. Yang, I. Collings, Z. Zhou, and E. Dutkiewicz, "Maximum flow-segment based channel assignment and routing in cognitive radio networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –6, may 2011.

[111] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*, EmNets '07, (New York, NY, USA), pp. 78–82, ACM, 2007.

[112] D. Cavalcanti, D. Agrawal, C. Cordeiro, B. Xie, and A. Kumar, "Issues in integrating cellular networks wlans, and manets: a futuristic heterogeneous

wireless network," *Wireless Communications, IEEE*, vol. 12, pp. 30 – 41, june 2005.

[113] R. Tomasi, H. Khaleel, F. Penna, C. Pastrone, R. Garello, and M. A. Spirito, "Frequency agility in IPv6-based wireless personal area networks (6LoWPAN)," in *WWIC'10*, pp. 146–157, 2010.

[114] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks." IETF RFC 6550, March 2012.

[115] K. Heurtefeux and H. Menouar, "Experimental evaluation of a routing protocol for wireless sensor networks: Rpl under study," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pp. 1–4, 2013.

[116] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, (Berkeley, CA, USA), pp. 2–2, USENIX Association, 2004.

[117] V. M. Bauer, "Routing in wireless sensor networks - an experimental evaluation of rpl, master's thesis," Master's thesis, Technische Universitt Mnchen, Ludwig-Maximilian-Universitt Mnchen, Universitt Augsburg, 2010.

[118] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks." IETF RFC 6551, March 2012.

[119] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "Rpl: The ip routing protocol designed for low power and lossy networks," tech. rep., Internet Protocol for Smart Objects (IPSO) Alliance, 2011.

[120] "IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1 –225, 16 2012.

[121] P. Du and G. Roussos, "Adaptive communication techniques for the internet of things," *Journal of Sensor and Actuator Networks*, vol. 2, pp. 122–155, 2013.

[122] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the Third Workshop on Embedded Networked Sensors*, 2006.

[123] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using RSSI value for distance estimation in wireless sensor networks based on ZigBee," in *Systems, Signals and Image Processing, 2008. 15th International Conference on*, pp. 303–306, June.

[124] M. Rondinone, J. Ansari, J. Riihijärvi, and P. Mähönen, "Designing a reliable and stable link quality metric for wireless sensor networks," in *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pp. 6–10, 2008.

[125] IAR Systems - World leading provider of software development tools for embedded systems, IAR System. [online]. Accessed: 15 May 2012.

[126] D. S. J. D. Couto, *High-throughput routing for multi-hop wireless networks*. PhD thesis, MIT, June 2004.

[127] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[128] S. Makridakis, S. Wheelwright, and R. Hyndman, *Forecasting Methods And Applications, 3Rd Ed.* Wiley India Pvt. Limited, 2008.

[129] C. S. Hilas, S. K. Goudos, and J. N. Sahalos, "Seasonal decomposition and forecasting of telecommunication data: A comparative case study," *Technological Forecasting and Social Change*, vol. 73, no. 5, pp. 495 – 509, 2006.

[130] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. http://otexts.com/fpp/, 2012.

[131] E. S. Gardner and D. G. Dannenbring, "Forecasting with exponential smoothing: Some guidelines for model selection," *Decision Sciences*, vol. 11, no. 2, pp. 370–383, 1980.

[132] O. Y. Ozalt, B. Hunsaker, and A. J. Schaefer, "Predicting the solution time of branch-and-bound algorithms for mixed-integer programs," *INFORMS J. on Computing*, vol. 23, pp. 392–403, July 2011.

[133] J. Bermudez, J. Segura, and E. Vercher, "Holtwinters forecasting: An alternative formulation applied to uk air passenger data," *Journal of Applied Statistics*, vol. 34, no. 9, pp. 1075–1090, 2007.

[134] E. S. Gardner, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[135] V. Vij and R. Mehra, "Fpga based kalman filter for wireless sensor networks," *International Journal of Computer Technology and Applications*, vol. 02, pp. 155–159, 2011.

[136] R. J. Meinhold and N. D. Singpurwalla, "Understanding the kalman filter," *The American Statistician*, vol. 37, no. 2, pp. pp. 123–127, 1983.

[137] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., Chapel Hill, NC, USA, 1995.

[138] E. Chan, *Quantitative Trading: How to Build Your Own Algorithmic Trading Business (Wiley Trading)*. Wiley, Nov. 2008.

[139] Amtel AT86RF231 Transceiver, Amtel. [online]. Accessed: 16 May 2012.

[140] X. Shen and S. Agrawal., "Kernel density estimation for an anomaly based intrusion detection system," in *The International Conference on Modeling, Simulation and Visualization Methods (MLMTA 06)*, 2006.

[141] M. Zoumboulakis and G. Roussos, "Integer-based optimisations for resource-constrained sensor platforms," in *S-CUBE'09*, pp. 144–157, 2009.

[142] P.-O. Gutman and M. Velger, "Tracking targets with unknown process noise variance using adaptive kalman filtering," in *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pp. 869 –874 vol.1, dec 1988.

[143] B. Kerkez, T. Watteyne, M. Magliocco, S. Glaser, and K. Pister, "Feasibility analysis of controller design for adaptive channel hopping," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, (ICST, Brussels, Belgium, Belgium), pp. 76:1–76:6, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[144] N. Armi, N. Saad, and M. Arshad, "Hard decision fusion based cooperative spectrum sensing in cognitive radio system," *ITB Journal of Information and Communication Technology*, vol. 3, pp. 109 – 122, 2009.

[145] The OpenWSN project, Confluence. [online]. Accessed: 14 April 2011.

[146] Wi-Spy Spectrum Analysis, MetaGeek Home of Wi-Spy and inSSIDer. [online]. Accessed: 12 December 2011.

[147] Contiki: The Open Source OS for the Internet of Things, Contiki. [online]. Accessed: 13 July 2013.

[148] TinyOS Home Page, TinyOS. [online]. Accessed: 15 March 2010.

[149] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of*

the 2003 ACM SIGMOD international conference on Management of data, SIGMOD '03, (New York, NY, USA), pp. 491–502, ACM, 2003.

[150] G. Wittenburg, N. Dziengel, C. Wartenburger, and J. Schiller, "A system for distributed event detection in wireless sensor networks," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, (New York, NY, USA), pp. 94–104, ACM, 2010.

[151] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, Aug. 2010.

[152] J. Burrell, "Disruptive effects of electromagnetic interference on communication and electronic systems," tech. rep., George Mason University, 2003.