Energy-Aware Decentralised Medium Access Control for Wireless Sensor Networks

George Christopher Smart

Communications & Information Systems Research Group
Department of Electronic and Electrical Engineering
University College London (UCL)

A thesis submitted for the degree of
Doctor of Philosophy (PhD)
in Electronic and Electrical Engineering

February 11, 2017

Statement of Originality

I, George C. Smart, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the text.

George C. Smart	February 11, 2017
	Date

Abstract

The success of future Internet-of-Things (IoT) based application deployments depends on the ability of wireless sensor platforms to sustain uninterrupted operation based on: (i) environmental energy harvesting and optimised coupling with the platform's energy consumption when processing and transmitting/receiving data; (ii) spontaneous adaptation to changes in the local network topology without requiring central coordination.

To address the first aspect, starting from practical deployments of a multi-transducer platform for photovoltaic and piezoelectric energy harvesting and the associated modelling and analysis, data-driven probability models are derived to facilitate the optimal coupling of energy production and consumption when processing and transmitting data. To address the second aspect (adaptability), the new concept of decentralised time-synchronised channel swapping (DT-SCS) is proposed – a novel protocol for the medium access control (MAC) layer of IEEE 802.15.4-based wireless sensor networks (WSNs). Simulation results reveal that DT-SCS comprises an excellent candidate for completely decentralised MAC layer coordination in WSNs by providing quick convergence to steady state, high bandwidth utilisation, high connectivity, robustness to interference and low energy consumption. Moreover, performance results via a Contiki-OS based deployment on TelosB motes reveal that DT-SCS comprises an excellent candidate for a decentralised multichannel MAC layer.

Publications

G. Smart, J. Atkinson, J. Mitchell, M. Rodrigues and Y. Andreopoulos, "Energy harvesting for the Internet-of-Things: Measurements and probability models," *2016 23rd International Conference on Telecommunications (ICT)*, Thessaloniki, 2016, pp. 1-6. (to appear).
doi: 10.1109/ICT.2016.7500416

G. Smart, N. Deligiannis, R. Surace, V. Loscri, G. Fortino and Y. Andreopoulos, "Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8538-8553, Oct. 2016.
doi: 10.1109/TVT.2015.2509861

N. Deligiannis, J. F. C. Mota, G. Smart and Y. Andreopoulos, "Fast Desynchronization for Decentralized Multichannel Medium Access Control," in *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3336-3349, Sept. 2015.
doi: 10.1109/TCOMM.2015.2455036

P. N. Whatmough, G. Smart, S. Das, Y. Andreopoulos and D. M. Bull, "A 0.6V all-digital body-coupled wakeup transceiver for IoT applications," *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Kyoto, 2015, pp. C98-C99.
doi: 10.1109/VLSIC.2015.7231338

N. Deligiannis, J. F. C. Mota, G. Smart and Y. Andreopoulos, "Decentralized multichannel medium access control: viewing desynchronization as a convex optimization method," *2015 14th International Conference on Information Processing in Sensor Networks (IPSN '15)*, pp. 13-24.
doi: 10.1145/2737095.2737108

G. Smart, N. Deligiannis, J. F. C. Mota and Y. Andreopoulos, “Demonstration Abstract: Decentralized time-synchronized channel swapping,” *2015 14th International Conference on Information Processing in Sensor Networks (IPSN ’15)*, pp. 404-405.
doi: 10.1145/2737095.2742557

G. Smart, R. Surace, N. Deligiannis, V. Loscri, G. Fortino and Y. Andreopoulos, "Poster Abstract: Decentralized Time-Synchronized Channel Swapping for Wireless Sensor Networks," *2014 11th European Conference (EWSN ’14)*.
doi: 10.1007/978-3-319-04651-8

H. Besbes, G. Smart, D. Buranapanichkit, C. Kloukinas and Y. Andreopoulos, "Analytic Conditions for Energy Neutrality in Uniformly-Formed Wireless Sensor Networks," in *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 4916-4931, Oct. 2013.
doi: 10.1109/TWC.2013.092013.121649

Patents

P. N. Whatmough, G. Smart, S. Das and D. M. Bull, "Communications device and method", Jun. 2015. Patent applied for. UK Patent Application, reference GB1509470.9.

Acknowledgements

It is hard to express with words, my gratitude for the guidance, structure and support that my supervisor, Dr Yiannis Andreopoulos, has imparted on me throughout the course of my PhD. For this, for his vast knowledge and varied skills, the "motivational speeches", and the endless ideas and opportunities (and endless patience), I will be forever indebted!

A huge thank you to Dr Nikos Deligiannis for his help, and Dr Ryan Grammenos for his excellent listening skills and moral support when I was learning the ropes (and whenever times got hard). Thanks to Dr Dujdow Buranapanichkit for showing me the aforementioned ropes in the first place and to Miss Hana Besbes for her mathematical input. Thanks to Prof Izzat Darwazeh for his guidance and feedback.

My thanks extend to all of my colleagues within the Communications & Information Systems Group at University College London (UCL), particularly those in R804 and MPEB7.06 labs, without whom the experience would have been much less enjoyable. A special mention to Miss Veronika Yordanova for the stimulating discussions and life advice over coffee. Thanks to my thesis examining committee, Dr Miguel Rio and Dr Mohammed Shikh-Bahaei, for a pleasant discussion.

My academic lifestyle would not have been possible without the understanding of some of my closest friends (listed alphabetically to avoid later arguments): Miss Jessica Baker, Dr Aaron Brown, Mrs Jessica Cooper, Mrs Jane Davies, Mr Christopher Hatton, Miss Charlotte Hossack, Miss Rhea Leahy, Mr Lorenzo Levrini, Dr David Mills, Miss Emma Mountford, Miss Ashleigh Paul and Mrs Nicola Phillips, all of whom tolerated my odd social hours and provided welcome distractions. I owe you all an incredible debt of gratitude.

Thanks to my parents, Mrs Linda Smart and Mr Christopher Smart. Without their extraordinary support and endless patience, it would never be possible to achieve my dreams! Encouraging the young engineer in me, even when that meant their possessions being taken apart. Thanks for the love, the moral (and never-ending financial) support, for keeping a roof above my head, and for ensuring I was always "fed and watered". — Yes, I have *finally* finished.

Thanks to Miss Irene Smart for the unwavering support during my writing up. The hospi-

tality and somewhere to rest my tired head was greatly appreciated. To Mrs Grace Smart, who unfortunately never got to hear the end of this story.

Lastly, to my sister, Miss Alice Smart. Your faith and encouragement when times got tough were *always* very much appreciated – your smile never fails to brighten my mood – you are my soul-mate for life! This thesis is dedicated you, Alice, for believing in me, even when I wasn't so convinced...

"You can kid the world, but not your sister."

— Charlotte Gray

Beyond all of those mentioned above, I am grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) and the UCL Electronic & Electrical Engineering department for funding this work.

This work was carried out between September 2011 and January 2016 in the Department of Electronic & Electrical Engineering at University College London (UCL).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet of Things (IoT) concept, that is, data connectivity between physical devices, vehicles, buildings and infrastructure, has recently acquired the commercial backing of major industrial stakeholders in the areas of networking, computer infrastructure and low-end wireless device manufacturers, ranging from wireless sensors to smart phones [5, 6, 7, 8, 9, 10]. This will result in millions of interconnected devices of varying size and technology collecting a wide range of information from the device's surrounding environment and uploading this data over an IP-based Internet connection for processing and analysis in the cloud [11, 12, 13]. Wireless sensor networks (WSNs) provide a platform to enable computational systems to gather data about every aspect of the physical world, and as a result, are an invaluable resource for realising the IoT. IoT is expected to allow for improved surveillance and monitoring, leading to smart homes and cities offering improved health care and transportation [9, 14, 13]. Robotics and manufacturing will also benefit from improved monitoring and reactive infrastructures [15]. The beginnings of this trend are already visible with smart electricity meters in the home allowing occupants to view energy demands in real time [16], and modern smart phone devices reporting on live traffic for potential routes. In many cities, waiting times for buses are reported at bus stops and at major transport hubs, as well as being available on smart phones, since buses are monitored as they travel throughout the city [17]. The availability of parking spaces across cities can often be seen on main roads entering a district, allowing for traffic to be routed away from busy zones, with guidance often extending directly to a vacant parking bay [18]. Aircraft engines are able to report on real time performance and maintenance requirements, such as temperatures, pressures and vibrations, preventing potential failures [15], while farming equipment equipped with smart sensors can report on the ideal time and location to plant or harvest a crop, sensing soil acidity, water salinity and temperature [12]. The information collected via wireless sensor networks has additional benefits in the field

of big data, allowing for the analysis of technological, business or marketing trends, such as creating alerts or analysing failure rates, recognising usage and activation patterns, establishing user or deployment-specific settings that are most often used, generating reliability estimates, calculating energy-bandwidth-computing costs, and so on. All of these scenarios require real-time sensor data, typically from the physical world, which can be provided by WSNs.

As the demand for machine to machine (M2M) connectivity continues, the Internet will have to change to incorporate billions of new devices [19], and so the need for systems to be self maintaining will grow, requiring networks to be able to initialise and configure themselves optimally, while managing their own energy supply requirements. Much of the data gathering in IoT systems is performed at the furthest points out on the network, the leaf nodes. Groups of leaf nodes may be configured to create a WSN, allowing nodes in radio proximity to communicate directly with each other at the outer edges of the IoT, where the physical interaction occurs. A border router is used to bridge the WSN to an IP-based network allowing sensors to report their observations back to cloud servers in large data centres for analysis [11, 12].

The WSNs themselves usually consist of small low power nodes, such as the the Crossbow TelosB mote [20] (also known as the Berkeley Tmote Sky), organised to form a collaborative network – that is, nodes with a common application goal while minimising self-inflicted interference. Figure 1.1 shows a TelosB node. Nodes may serve to monitor physical or environmental conditions with application specific sensors. For autonomous networks running with limited energy resources (e.g., batteries), one of the key metrics is network lifetime [21]. To combat this, future WSN deployment infrastructures are expected to be equipped with energy harvesters (e.g., piezoelectric, thermal, photovoltaic), to help replenish the node battery where possible. By matching the amount of energy expected to be harvested from the environment over a given time interval (e.g., 24 hours in the case of solar harvesting) with the expected energy dissipation for the same period, it can be said that the sensor node achieves energy neutrality [22]. That is, the node is expected to be able to operate in perpetuity without the requirement for human intervention.

At the core of any sensor node or IoT device is a microcontroller unit (MCU), usually a small system on a chip (SoC) computer, which performs all of the computing functionality required by the node. A radio transceiver is used to communicate with other nodes via RF, either across *(i)* local area capillary networks (between leaf nodes), such as IEEE 802.11 [23] or IEEE 802.15.4 [24], or via *(ii)* wide area cellular networks such as the enhanced machine-type communication (eMTC) [25] detailed in Release-13 of the 3GPP specification [26], or the LoRa Alliance LoRaWAN [27]. The node will usually include an omni-directional antenna, commonly a planar inverted-F antenna (PIFA), as used on the TelosB. The node sensing hardware is

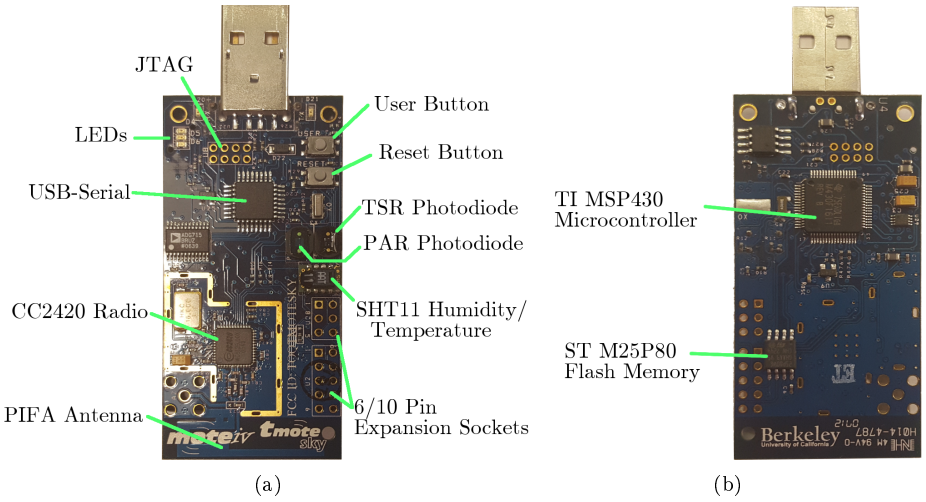


Figure 1.1: Annotated components of TelosB/Tmote Sky node top side (a) and bottom side (b) with battery compartment removed.

dependant on the measurement task at hand, typically low cost sensors are selected allowing the nodes to be deployed in bulk. Non-volatile flash memory is often used to store sensing parameters and configuration data for the node, as well as holding sensor measurements until they are ready to be transmitted.

Of all the node components, the radio transceiver consumes the most energy [21, 28, 29, 20] and has the biggest impact on energy use, substantially reducing the node operating time if used suboptimally. Since the MAC layer is directly responsible for controlling the radio transceiver, it is important to use a suitable MAC layer for each deployment.

As the energy available to a node is dependent on the battery capacity and charge level, as well as the environmental conditions for energy harvesting, it is important to maximise energy efficiency when sensing, processing and transmitting data, together with understanding and predicting the performance of energy harvesting schemes.

1.1 Aim and Scope

Early WSN and IoT capillary networks covering a local area [often referred to as wireless personal area network (WPAN) or wireless local area networks (WLAN)] were typically built upon the IEEE 802.15.4 [24] or IEEE 802.11 [23] MAC layers and usually required an interface to IP-based networks, such as the Internet. More recently, the 3rd Generation Partnership Project (3GPP) Workplan Release-13 [26] have proposed enhanced machine-type communication (eMTC) protocol [25] as well as the preexisting LoRaWAN [27]. Both protocols can co-locate with existing cellular network infrastructure, enabling sites to better support machine-to-machine (M2M) communications, and are designed to fulfil the requirements of wide area

networks commonly envisaged with IoT. Such a network is typically referred to as a low-power wide-area network (LPWAN). These LPWAN networks are optimised for geographic cell area coverage and node battery life, as opposed to traditional cellular networks which are optimised for ultra high bandwidth with a very expensive energy cost through network protocol overheads.

Both IEEE 802.15.4 [24] and IEEE 802.11 [23] MAC layers support carrier sense multiple access with collision avoidance (CSMA/CA) at the MAC and physical (PHY) layer. The carrier sense multiple access (CSMA) part of the protocol implies that nodes first listen prior to transmitting to ascertain if the medium (channel) is already in use. This is referred to as performing a clear channel assessment (CCA). The collision avoidance (CA) part of the protocol states that if activity is present, the transmitting device must wait a typically random (to avoid repeated, reoccurring collisions) *backoff* time, before performing a another CCA and trying the transmission again.

Although IoT and WSN nodes do not require bandwidth in the same order of magnitude as has become available on smart phones, CSMA/CA falls short of optimum with even these modest bandwidth and energy requirements. This is because transmission opportunities are wasted due to the CSMA/CA backoff time (the time a node must wait idly to reduce network congestion when the channel is busy), while energy is wasted when repeatedly checking to see if the radio channel is clear (the CCA). Furthermore, CSMA/CA cannot guarantee that all self inflicted interference (i.e., caused by nodes in the WSN network itself) and collisions are avoided [30], and so more energy is wasted in retransmissions due to lost data. This is further elaborated on in Section 2.3.1. It is clear that a better solution, without packet collisions and CCA is required in order to fully utilise the bandwidth available for communication in a densely populated WSN. Time division multiple access (TDMA) offers collision free access to the wireless medium, and is one of the best schemes for this purpose, supporting both centralised and decentralised configurations. Given that many WSN deployments often do not rely on a single central (coordination) node, a decentralised deployment of TDMA may be best suited.

Chapter 3 begins by assessing what is possible with existing transducer technologies, deploying a small multi-transducer platform to collect harvested energy data from commonly available types of energy harvester in several real-world environments. The focus is on providing measurements and raw data from the platform, as well as associated tools to capture new data. Raw data is parsed to create statistical models for each harvesting technology in each of the deployed locations. The data produced gives a system designer practical values to use in modelling frameworks, such as presented in Chapter 4.

Chapter 4 investigates energy management and looks to minimise the energy requirement of nodes by optimising the number of nodes in cluster-trees to match the expected data rate.

It takes TDFMA [31], a MAC protocol similar to the DT-SCS protocol described in Chapter 5, and derives an analytic framework for achieving energy neutrality in uniformly-formed WSNs. The framework allows a system designer to understand the link between network parameters (i.e., number of nodes, timeslot length, topology, duty cycle, and so on) and the associated energy requirements of the node influencing the battery selection and energy harvester.

Chapter 5 proposes a practical low-energy distributed multichannel TDMA MAC-layer coordination protocol for use in decentralised IoT and WSN deployments, ensuring good energy efficiency and bandwidth utilisation, with quick convergence to the steady state, and no requirement for centralised time synchronisation or a coordinating node.

Throughout this thesis, the main focus is on maximising node lifetime through: *(i)* a realistic understanding of expected energy harvesting, *(ii)* the careful tuning of network parameters, and *(iii)* a specifically designed MAC layer which not only meets low energy requirements, but also offers low latency, quick convergence to a steady state and high bandwidth utilisation.

1.2 Thesis Structure & Contributions

In light of what is discussed above, this work's contributions are outlined in list provided below:

1. Chapter 3 details an experimental study of harvesting energy from readily available harvesting transducer technologies currently available. A purpose-built multi-transducer harvesting platform is deployed in several indoor and outdoor scenarios, collecting raw data for harvested power from photovoltaic and piezoelectric sources. The generated power profiles are coupled with probability mixture models to create data driven probability models that characterise the energy harvesting process for each transducer in the given scenarios. Specific contributions are:

   (a) Development of a purpose built multi-transducer energy harvesting platform enabling real time logging of produced energy to SD memory card.

   (b) Tools for processing the recorded data to eliminate idle periods, generation of energy histograms and theoretical probability distributions for harvested energy.

2. Chapter 4 investigates the perpetual operation of a sensor node by balancing the node's expected energy consumption with its expected energy harvesting capability to operate a network in energy autonomy. Conditions for energy neutrality are derived assuming a uniformly-formed WSN parametric to: *(i)* the duty cycle for the network activation; *(ii)* the number of nodes in the same tier of the cluster-tree topology; *(iii)* the consumption rate of the sink node(s) that collect (and possibly relay) all sensor measurements; *(iv)*

the marginal PDF characterising the data transmission rate per sensor node; *(v)* the expected amount of energy harvested by each sensor node. This resulted in the following contributions:

(a) For each tier of a WSN cluster-tree topology, analytic derivation of the number of nodes that leads to the minimum requirement for harvested energy under four commonly encountered marginal probability density functions (PDFs) for the data transmission rate per sensor.

(b) Analytic comparison of the minimum requirements for energy harvesting under different application parameters and different data transmission rates.

(c) Validation of the theoretical results via an energy measurement testbed using TelosB sensor nodes employing the recently proposed collision free protocol (TFDMA [31]).

(d) Establishment of the optimal operational parameters within two application scenarios for WSN-based monitoring and data collection.

3. Chapter 5 proposes decentralised time-synchronised channel swapping (DT-SCS), a novel self-managed, decentralised, collision free, protocol for the MAC layer of capillary networks such as IEEE 802.15.4, commonly used with WSNs. By using negatively-coupled pulsed coupled oscillators (PCOs) within each channel (a.k.a. DESYNC [32]) to create a TDMA-style schedule within each channel, and positively-coupled with PCOs (a.k.a. SYNC [33]) to align the TDMA schedules across multiple channels, the proposed protocol ensures wireless nodes converge to synchronous beacon packet transmissions across all 16 IEEE 802.15.4 channels with a balanced number of nodes in each channel without the need for global time synchronisation. Peer-to-peer channel swapping is possible through swap requests and acknowledgements made by nodes in neighbouring channels, allowing node pairs to switch channels without disrupting the stability or bandwidth efficiency of the network. Spontaneous adaption to available transmission slots is achieved by using an elastic (rather than rigid) time synchronisation method and by configurable coupling coefficients for SYNC and DESYNC. Extensive comparisons between DT-SCS and TSCH are carried out in terms of convergence time, bandwidth utilisation, connectivity and robustness to packet losses via simulations and a hardware testbed. Beyond the implementation of the protocol, the thesis makes the following contributions:

(a) Proof that DT-SCS converges to a balanced steady-state and estimation of the expected connectivity and energy consumption.

(b) Detailed simulation results demonstrating the efficacy of the proposed protocol for distributed multichannel coordination in WSNs.

(c) Detailed comparisons between DT-SCS and TSCH in terms of energy consumption, convergence delay, bandwidth efficiency, robustness to interference and the existence of hidden terminals and node churn, both in a event driven simulator written in MATLAB and using TelosB sensor nodes.

To conclude, Chapter 6 presents a summary of the work undertaken and highlights some future potential directions.

Chapter 2

Background

Given that the thesis contributions are in three separate areas, the literature review is separated into three separate sections. The first part of the literature review discusses common energy harvesting transducer technologies, as well as the harvesting circuitry required to optimise energy scavenging. The second section of the literature review looks at energy management policies and how they can inform the network design to operate most efficiently. Finally, the literature review looks at the design and limitations of existing MAC protocols.

2.1 Energy Harvesting

The availability of energy sources greatly dictates the choice of harvester technology. Essentially, the available sources of energy are: light, radio-frequency (RF), electromagnetic radiation, thermal gradients and motion (including fluid flow). Ambient RF, electromagnetic radiation and thermal gradients have received some attention (e.g., the Seiko thermic watch), but the availability of significant power levels with these technologies is an issue, and, for the case of RF, efficient extraction using devices much smaller than the radiation wavelength is key challenge [34]. Beyond these energy sources, fuel based generation using ambient fluids, such as human bodily fluids, has also been reported [34]. While RF energy scavenging from television transmitters, Wi-Fi access points and GSM base-stations may be a viable option for devices in urban environments, these energy sources are less prevalent in rural areas. Thermal harvesting (e.g., the Seebeck effect) has the potential to generate ample power, but, in practice the physical configuration required to maintain a large enough thermal gradient can often be hard to achieve. Electromagnetic harvesting requires the availability of stray magnetic fields, and therefore is only suited to limited deployments. Overall, the general consensus from the related literature [35, 36, 34, 37, 38, 39, 40, 41, 42] is that piezoelectric and photovoltaic energy transducers are

Table 2.1: Energy Harvesting Sources

Technology	Power Density
Photolvoltaic	15mW/cm^3 [37]
Radio Frequency	$7\mu W/cm^3$ [40]
Electromagnetic	$23\mu W/cm^3$ [41]
Thermal	$1-10\text{mW/cm}^3$ [42]
Piezoelectric	$330\mu\text{W/cm}^3$ [39]

the most versatile, commercially mature technologies to consider for WSN and IoT-oriented deployments.

The literature on energy harvesting approaches for wireless sensors and IoT oriented platforms can broadly be separated in three categories. The first category relates to physical properties and design of transducer technologies that scavenge energy from the environment. Table 2.1 summarises the typical power densities from these sources. The focus of existing research work in this category is on the physical design of harvesters, in maximising transducing rates [43, 44, 40] and the efficiency of associated power conditioning circuitry [45, 46, 47], which must match the input impedance of the harvester circuitry to the (continually changing) output impedance of the harvester transducer [48] in order to maximise energy extraction, rather than concentrating on the statistical characterisation of the manner with which energy is converted across time and within different environmental conditions. While there is already a body of work on statistical characterisation of node energy consumption in several application domains [49, 50, 39, 51, 52, 53], very few data driven statistical characterisations of energy harvesting exist, requiring more experimental evidence from practical testbeds.

Energy harvesters are customarily combined with a method of storing the harvested energy to allow for continued operation when the energy source is not available. One such example is where an array of photovoltaic cells are used to power a sensor node, and a super-capacitor or battery is used to allow operation during short time intervals caused by cloud cover, or longer intervals such as at night. Harvested energy in excess of the sensor node's immediate requirement is stored for later use, and consumed when the harvested energy alone is insufficient to maintain sensor operation.

The task of controlling power flow between harvester, sensor node and storage device, as well as tracking the maximum power point, is commonly handled by a power controller IC.

The last category of literature on harvesting relates to energy management frameworks, which is discussed in Section 2.2 below.

2.2 Optimisation of Energy Consumption in WSNs

In WSNs, dynamic power management (DPM) may be used to minimise energy consumption in a sensor node or network, but it must have minimal impact upon the wireless networks performance metrics. This enables a greater active operating period for a fixed sized energy source (e.g., battery), or reduces the energy harvesting requirements (e.g., smaller solar panels) in an energy scavenging scenario. Since DPM is a vast subject, only fundamental principles related to the development and validation of energy aware MAC protocols are discussed here.

It is common sense that a node within a network should return to the lowest energy state possible (typically a 'sleep' state) when there is no work to be done, in order to minimise energy use [54]. Modern IoT sensor operating systems, such as Contiki-OS, have inbuilt support for DPM, and power awareness is at the core of the OS development. Contiki-OS includes energy estimators [55, 56] which can help track where nodes are using energy and aid the developer in writing more energy efficient algorithms and implementations. Features of these modern real-time operating systems include automatically powering down the radio transceiver chipset and/or microcontroller when the OS detects either is not in use. Other operating systems such as TinyOS require the programmer to explicitly power down parts of the node hardware in code when they are not in use, and then re-power them before use.

As is discussed to greater depth in Section 2.3, several MAC layer protocols attempt to minimise their energy consumption. Drawing from the considerable research literature on DPM, two representative cases are highlighted: *(i)* Pantazis *et al.* [57] propose a TDMA-based scheduling algorithm using sleep mode to save energy consumption; *(ii)* the monitoring of an industrial electric system by Salvadori *et al.* [58] used WSNs with DPM to achieve a very long operational lifespan. The reader is referred to the survey paper by Bachir *et al.* [21] for further examples of DPM-based approaches in WSNs.

Other research tackles the problem of DPM using approaches to achieve lower energy WSN-based monitoring [59, 60, 22]. Technology-oriented approaches design new circuits and systems for more efficient energy management [39, 61], while others strive for more efficient scheduling and transmission protocols [31, 3, 62, 63]. One thing, however, is common; they all try to bridge the gap between the data sensing and transmission requirements, the corresponding energy production (e.g., via a harvesting unit), and energy storage capability of the underlying hardware. Finally, another group of approaches proposes optimal energy management policies under given energy harvesting, sensing and transmission capabilities [51, 64, 65, 66, 67]. Such policies optimise the manner each sensor node performs its data gathering and buffer management in order to minimise the required energy consumption.

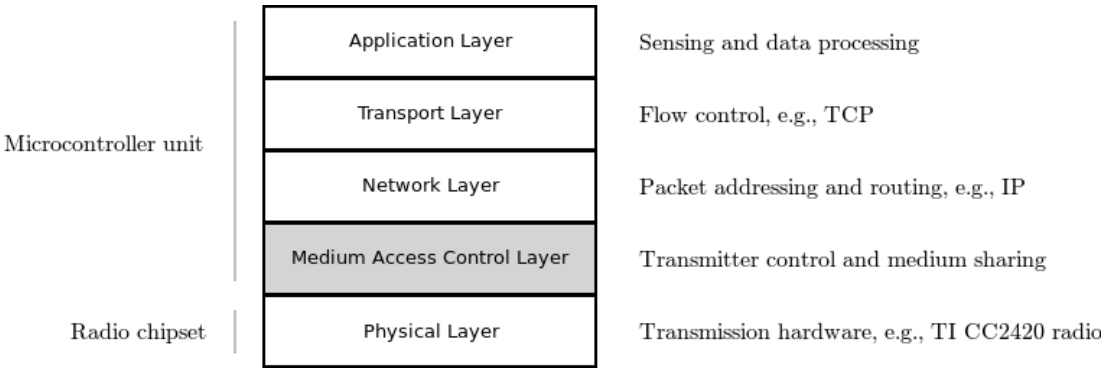


Figure 2.1: Conceptual illustration showing interfacing of key network layers, along with their function and location inside the node.

2.3 MAC Protocols

The PHY layer of the network specifies how the raw bitstream is sent via the physical channel medium. It is responsible for powering the transceiver, performing CCAs and setting the transmission frequency, as well as modulating data onto the RF carrier. In the case of the 2.4 GHz IEEE 802.15.4 PHY [24], direct sequence spread spectrum (DSSS) is used, employing offset quadrature phase-shift keying (O-QPSK) modulation, giving a channel bitrate of 250 kbps in 5 MHz bandwidth [29, 24]. The MAC layer is part of the data link layer and handles all access to the wireless channel. The MAC is responsible for providing links between nodes, handling beacons, controlling node association and disassociation, maintaining guaranteed time slots (GTS) or performing CSMA channel access [24]. The network and transport layers control the route data packets take through the network. These layers need not be aware of the underlying MAC and PHY layers; they send data generated by the application layer above, such as sensor data. Figure 2.1 shows the order of the layers for WSN nodes.

Typically, the PHY layer is fixed by the standard and chipset manufacturer, so it is not possible to change the PHY behaviour significantly. This maintains compatibility with existing hardware. The higher layers are implemented as software inside the node's MCU. As the MAC layer controls access to the wireless medium via the radio and is responsible for powering the transceiver, node association/disassociation and data links, it allows for careful manipulation of node energy expenditure.

Within the scope of WSNs, nodes have limited computational power and small amounts of memory, as well as a limited resources for maintaining timing and synchronisation. As mentioned, the MAC protocol should offer access to the underlying wireless channel in an energy efficient manner, while maintaining high reliability, high bandwidth and low latency despite many hardware limitations.

As discussed above, it is generally accepted that the main use of energy in capillary WSN

Table 2.2: Typical power values of Crossbow TelosB motes for different activity states. Power values are reported in milli-Watt (mW). Transmitter reported at full power (0 dBm).

MSP430 MCU		TI/Chipcon CC2420 Radio chip			
Sleep	Active	Sleep	Idle	Receive	Transmit
0.0153	5.4	0.003	1.28	59.1	52.2

and IoT deployments is the radio transceiver [21, 28]. Table 2.2 presents the power requirements for a Crossbow TelosB mote operating on 3 Volts in different activity states [29, 20]. Evidently, receiving and transmitting incurs more than an order of magnitude higher power dissipation than the remaining operational modes of the MCU and the radio chip. Therefore, given the energy expense of having the receiver on, minimising the time the receiver is powered is important. Common losses in energy efficiency at the MAC layer stem from:

- **Idle Listening** — the undesirable circumstance of having the receiver enabled although no node within range is sending data.
- **Overhearing** — when a node invests energy in the reception of frames which are not relevant to it, for example, unicast (one-to-one) packets addressed to other nodes or long preambles used to synchronise nodes.
- **Collisions** — when a receiving node can hear more than one transmission at a time. In this case, the receiving node cannot resolve either of the two transmissions and so the energy for receiving is wasted. If no other nodes resolve the transmission, then the energy expenditure of both nodes for transmission may also be wasted.
- **Protocol Overhead** — extra data nodes are required to transmit and receive to maintain the network functionality, such as control signals or time coordination. This can have a large effect on energy dissipation when network frames are small, as the ratio of control data to payload data becomes significant.

Two main schemes exist when regulating a shared wireless medium; contention-based and reservation-based.

2.3.1 Contention Based MAC Protocols

Contention based protocols do not require *a-priori* knowledge of the network or global node time synchronisation and are therefore a first choice in simple low traffic networks because of their relative ease of implementation. The additive links on-line Hawaii area network (ALOHA) protocol and CSMA are examples of contention based protocols. Very simple protocols, such as pure (or unslotted) ALOHA [68] require nodes to send data as soon as they have it (without

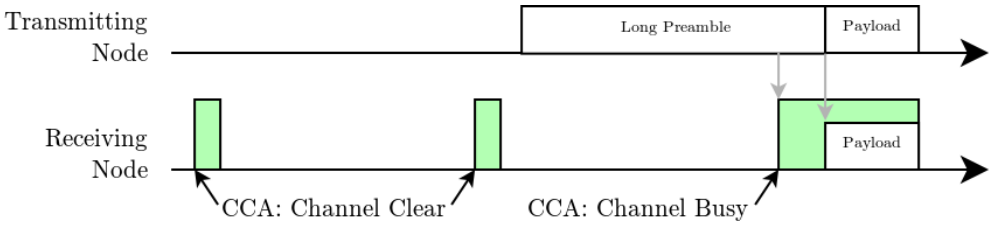


Figure 2.2: With preamble sampling, nodes periodically wake from sleep to check for radio activity. If activity is detected, the node stays awake to receive the next packet. To send a packet, the sender may repeatedly send a preamble or the same data packet until the duration of the channel check interval (time between CCAs) has elapsed.

performing any CCA), and wait for an acknowledgement from the receiving node confirming error free reception. If after a predetermined time period no acknowledgement is received, the sending node re-transmits the data after a random delay. Using pure-ALOHA, the maximum usable bandwidth of a channel is approximately 18%, following the standard discussion presented by Tanenbaum *et al.* [69], with all frames of equal length, infinite retries, and population of transmitting stations following a Poisson distribution. Slotted-ALOHA [70] is an improvement on pure-ALOHA which introduces timeslots. A node may only transmit within timeslots, and thus, the number of collisions is reduced. The maximum usable bandwidth of the channel rises to approximately 37% [69] under the same conditions as before, but nodes also require global time synchronisation.

With CSMA, a node deciding to transmit first performs a CCA to see if the channel is in use. If the channel is clear, the node is free to transmit. If not, the node must wait a random *backoff* period before reattempting the transmission again.

Since there is no time synchronisation (i.e., nodes can transmit at any time), a receiving node cannot predict the transmission of data. As the cost of idle listening is large, efforts must be made to minimise the time the receiver is on. The most common approach used in WSNs is *preamble sampling*, shown in Figure 2.2. Here, receiving nodes perform a CCA by periodically (every few hundred milliseconds) sampling the channel for a short time (several hundred microseconds) to determine if a transmission is ongoing. The receiver duty cycle, that is, the average amount of time the receiver is on for, is low. Nodes sending data transmit a preamble at least as long as the receiver's CCA sampling interval; this ensures that the all receivers are listening to the data which follows but allows the receiver to be powered down for the majority of the time when no channel activity is present [71, 72].

As contention based protocols do not incorporate heavy network overheads and do not require topographic knowledge or global time synchronisation, they are good candidates for networks supporting low traffic volumes with a low node density and where there is limited (or no) time synchronisation between transmitter and receiver. Occasionally scheduling is not

possible, such as with the AX.25 packet radio protocol [73] used to transmit telemetry data globally between millions of radio amateurs. Performance of contention based network protocols under heavy volumes of traffic is poor due to collision backoff periods and retransmissions, as discussed previously. As such, contention based protocol networks cannot obtain the same efficiency as ideal reservation protocols [1]. Adding lengthy preambles increases the energy consumption of the transmitter through the transmission of more data (protocol overhead) and increases receiver energy use through overhearing, as well as reducing network bandwidth.

2.3.2 Reservation Based MAC Protocols

Reservation based protocols create a schedule which reserves a *timeslot* for each node to transmit in a channel. This requires time synchronisation between all nodes to work effectively, since nodes must know at what time they are allowed to transmit. This gives rise to the concept of a *global clock*. Commonly, a central *coordinator* node creates a schedule with certain objectives (e.g., maximum network bandwidth, minimum latency, maximum energy efficiency, fairness in the transmission opportunities provided to each participating node) as well as maintaining time synchronisation. Creating this schedule requires knowledge of the network topology, which adds further complexity. TDMA and the newly standardised time synchronized channel hopping (TSCH) are common examples of a reservation based protocols, where time is divided into slots [1]. These slots are grouped together into frames that repeat over time. The schedule may specify slots where a node is to transmit or receive: if more than one node is allowed to transmit per slot, access is controlled by a contention based scheme within the slot. Conversely, the schedule may specify a contention free period (slot) during which time, only a single node is permitted to transmit to a single receiver. If the schedule is correctly designed then these transmissions will not suffer contention or collisions [31, 3, 62, 63, 74] which ensures calculable packet latency and increased bandwidth in high traffic networks with fairness amongst nodes.

Studies show that reservation based protocols clearly outperform their contention based counterparts in terms of bandwidth efficiency, but at the potential cost of latency, increased overheads, topology knowledge and network time synchronisation [75, 76, 77, 69].

2.3.3 Channel Hopping

Within the context of WSNs and IoT, channel hopping enables nodes to move between the 16 channels of the industrial, scientific, and medical (ISM) radio band (2400-2484 MHz) as specified in the IEEE 802.15.4 PHY [24] or the 7 channels of the dedicated short range communication (DSRC) radio band (5850-5925 MHz) [78]. Expanding a node's schedule across multiple channels allows for increased network bandwidth and improves robustness to packet

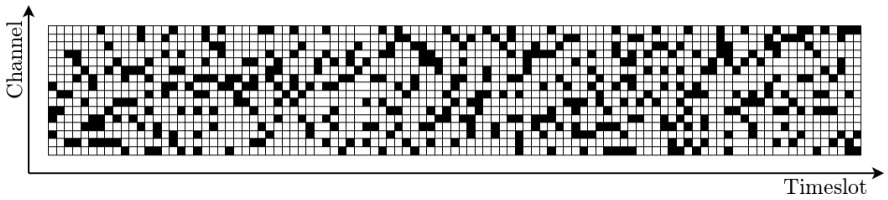


Figure 2.3: Example TSCH [1] schedule with 25 nodes, 101 timeslots and 16 channels derived by the 6tisch simulator [2]. Black cells indicate slots used for unicast connectivity, white cells are unused.

loss stemming from interference in these unlicensed bands [79, 80]. Channel hopping enables nodes to be evenly spread across (and move freely between) channels to help minimise packet loss, since nodes are not fixed in channels with high levels of interference. With TSCH [1] and multichannel DSRC [78] now comprising essential elements of the IEEE 802.15.4e-2012 [81, 82] and IEEE 802.11p [23] standards, respectively, the concept of channel hopping has gained acceptance as a good solution to avoid interference whilst maintaining high node connectivity and network throughput. Figure 2.3 shows a visual example of channel hopping from a TSCH schedule generated by the 6tisch simulator [2]. The schedule is a grid of cells with timeslots (time) along the x-axis and channel (frequency) along the y-axis. Black cells indicate unicast communication within a slot while white cells are unused timeslots. The channel hopping mechanism of TSCH is discussed further in Section 2.3.5, below.

2.3.4 Comparison of Existing MAC Solutions for WSNs

Multichannel MAC layer coordination may be achieved in a number of ways. The simpler approaches use schemes that assign channels to nodes in a *static* manner to balance them across available channels in a given radio band and maximise bandwidth use [83, 84, 31]. Such solutions, however, try to minimise the rate of channel hopping, as this affects network stability and tends to decrease the achieved transmission rate per node. As such, they achieve reduced node connectivity and are prone to persistent interference in any of the used channels caused by node churn (nodes joining or leaving a channel).

Other protocols implement *dynamic* coordination of node channel hopping throughout the lifetime of the WSN. Hwang *et al.* [85] proposed a low energy, receiver driven, channel hopping scheme for WSNs that does not require global time synchronisation. Instead, each sender predicts the wakeup time of each receiver encountered, which is shown to minimise idle listening at the cost of significantly reduced bandwidth efficiency. Tang *et al.* proposed Efficient Multichannel MAC (EM-MAC) [3], a multichannel protocol based on Receiver Initiated MAC (RI-MAC) [86] and Predictive Wakeup MAC (PW-MAC) [87]. In EM-MAC, nodes select the channel for communication by following pseudo-random scheduling, as in the predictive wakeup approach.

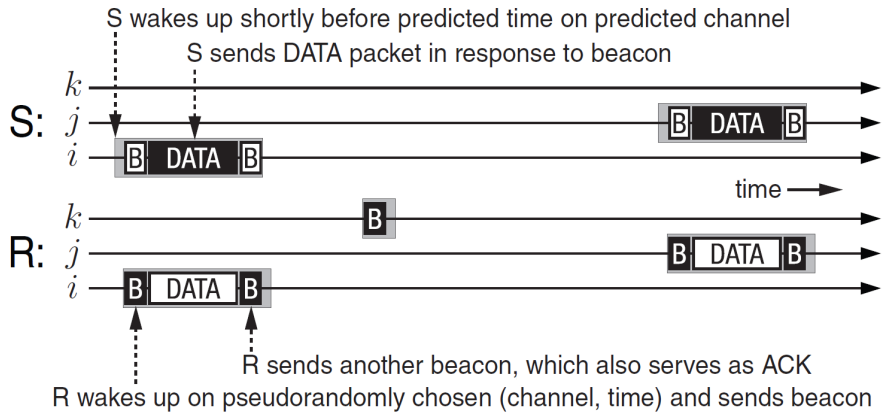


Figure 2.4: Data transmission between node S and node R under the receiver initiated EM-MAC, as presented by Tang *et al.* [3].

Since EM-MAC is receiver initiated, receiving nodes beacon according to the schedule – any node with data to send to the receiver is allowed to do so. The receiving node beacons after valid data reception to serve as an acknowledgement and initiate further data transfers. This process is shown in Figure 2.4. EM-MAC is shown to be highly resilient to interference and jamming with similar energy characteristics to Predictive Wakeup MAC, albeit at the cost of substantially higher duty cycle and low bandwidth efficiency.

Alternative approaches for multichannel coordination and channel hopping use a control (or "coordination") channel, where nodes negotiate another channel to use for data transmission. Representative examples include Y-MAC [4], A-MAC [88], MMAC [89], CAM-MAC [90], MuChMAC [91] and the TSCH [1] option of IEEE 802.15.4e-2012 [82, 81].

Y-MAC [4] uses a hybrid of contention and scheduling mechanisms for access control. Scheduling the receiver wakeup times helps to minimise idle listening and overhearing. A base station or "sync" node sends timing packets on a control channel to start the network. These packets also serve as a way to provide for time synchronisation. Transmitting nodes first compete during a contention-based broadcast (one-to-many) period to transmit to the receiver during a slot in the unicast period. Contention-based backoff methods are used within the broadcast section to schedule slots in the unicast section, as shown in Figure 2.5. Periodic control frames, containing time synchronisation data, are also sent so as to keep the broadcast–unicast frame structure aligned.

A-MAC [88] establishes an optimal timeout value for each node to periodically wake up so as to send and receive packets. In conjunction with a rate estimation scheme, A-MAC is shown to decrease energy consumption compared to previous approaches.

Mobile adaptive MAC (MMAC) [89] is designed as a protocol suitable for mobile nodes by using dynamic frame times to allow nodes to send data without long waiting periods, i.e., before the network topology or propagation changes. Transmission is contention free, with

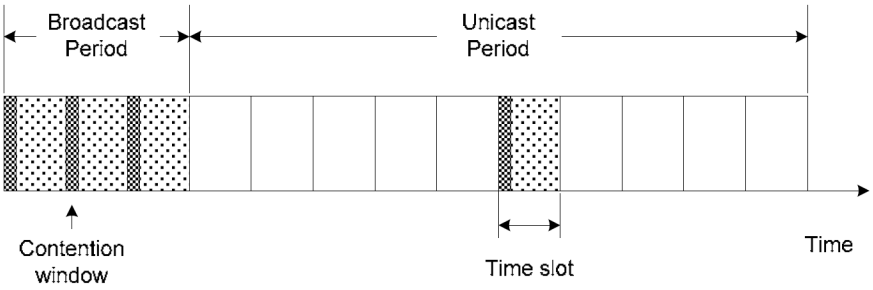


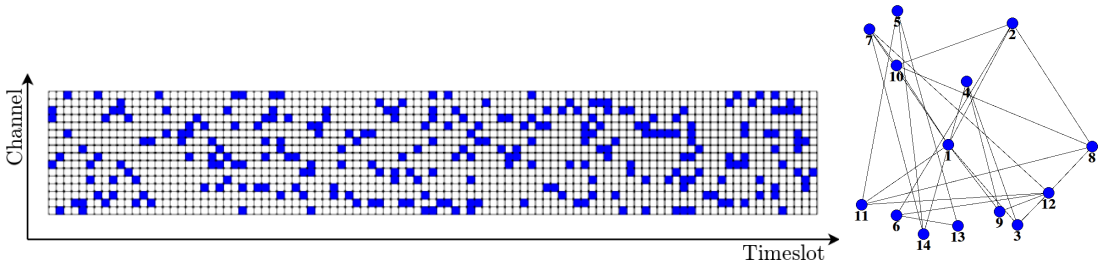
Figure 2.5: Frame architecture of Y-MAC, as presented by Kim *et al.* [4].

frame time calculated as a function of node mobility. Nodes are required to know their location and movement, which is a disadvantage since nodes must invest large amounts of energy to establish this information (e.g., via on-board GPS hardware).

Cooperative asynchronous multi-channel MAC (CAM-MAC) [90] uses *cooperation* at the centre of the protocol design. Nodes creating new connections are advised by neighbours as to which channels would cause the least disruption when not all neighbours can be heard. This is done by the transmission of probe packets announcing a node's intention to establish a new connection. Neighbouring nodes are then allowed to provide feedback to the probe in the form of an invalid response, indicating the connection is not in their interest. If the probe goes uncontested, then the connection is established, otherwise the probe fails and the sending node creates a new session elsewhere.

MuChMAC [91] is a low-overhead dynamic multichannel MAC for WSNs. The protocol was designed to be general purpose and suitable for a wide range of traffic rates. Energy efficiency is achieved by very low duty cycle (a few percent) and collisions are minimised by using subslots within each TDMA timeslot determined by the node ID. This gives performance similar to other multichannel TDMA protocols under high traffic load, while performance under light traffic is similar to single channel protocols.

Finally, time-synchronized channel hopping (TSCH) [1] is a frequency hopping reservation based protocol that uses a group of timeslots in multiple channels to make a frame, with nodes advertising for connections on a control channel, and then rendezvousing for data transfer on another channel. In comparison to previously described protocols, TSCH strikes a good balance between bandwidth utilisation, energy consumption and node connectivity. It was adopted as an optional mode within the IEEE 802.15.4e-2012 standard [28, 82, 81], and is currently developed via the open-source openWSN effort and the related 6tisch simulator for an associated IETF RFC [2] and therefore, can be considered as the definite benchmark for multichannel MAC protocols. For this reason, it's basic operating functionality is further described in Section 2.3.5, below.



(a) (b)

Figure 2.6: (a) TSCH slotframe structure and (b) corresponding connectivity mesh, showing 14 nodes within 16 channels and 101 timeslots, derived by the 6tisch simulator [2].

2.3.5 Time-Synchronised Channel Hopping (TSCH)

Figure 2.6(a) depicts an example of the TSCH protocol [1] schedule, where an arbitrary topology [Figure 2.6(b)] is formed between 14 nodes [2]. The protocol was designed for networks that contain mobile nodes and stemmed from previous work on the time-synchronized mesh protocol (TSMP) [92] protocol and the wirelessHART standardisation [92, 93]. Each node reserves timeslots within a rigid (predefined) slotframe interval [horizontal axis of Figure 2.6(a)] and within the 16 channels of IEEE 802.15.4 [vertical axis of Figure 2.6(a)]. Unoccupied slots appear in white. As the slotframe interval of Figure 2.6 repeats periodically, all nodes transmit and listen in different channels, thus avoiding concentrated interference.

TSCH employs a control channel where nodes should advertise or listen for connection requests during their idle slots, deciding whether to advertise or listen randomly, based on an advertising parameter. Nodes use an ALOHA style protocol on the control channel, simply transmitting their advert into a (randomly selected) timeslot. Each advert contains a selection of timeslots the transmitting node has free, and a randomly selected channel. Other nodes listening to the control channel during their own idle slots compare the received advertised slots with their own slots, and matching free slots are used to schedule a connection during the next slotframe. The receiving node responds to the advertisement during the matching slot, or a random matching slot if more than one exists. A similar (though not identical) structure applies for the IEEE 1609.4 multichannel DSRC extension [94, 78] of the IEEE 802.11p [23] standard, featuring advertisements within a rigid timeslot structure.

A rather complex advertising request and acknowledgement (RQ/ACK) process occurs [1, 82] over the next few slotframes, with the replying node issuing a transmit connect request, waiting for a acknowledgement and handshaking before the connection is ready for data transfer. This initial handshaking also uses the ALOHA protocol, and so when few free slots are available, there is a high chance of collision within the communication slot also, so it becomes progressively harder to fill the slotframe. Since the advertised channels are chosen at random, the timeslot-

channel-pair may also be in use by another node. The control channel is prone to interference and self-inflicted collisions when nodes are set to advertise slot reservations very aggressively. If slot advertising is not aggressive and nodes leave the network (e.g., in vehicular networks [95], or networks of mobile robots or drones), their slots may remain unoccupied for long periods of time, which limits the bandwidth use per channel.

This can be seen, for instance, in the large number of unoccupied slots in Figure 2.6, and ultimately means the network takes longer to converge to a steady predictable state, and that it does not make full use of available channel bandwidth. However, both high bandwidth and quick convergence are important for ad hoc networks that must quickly converge to a steady operational state and transmit high data volumes under a periodic or event driven schedule [96, 97, 98, 95].

TSCH [1] cannot be considered as an infrastructureless protocol since; *(i)* a coordinator node is required in order to maintain global time synchronisation via beacon message broadcasts at slotframe boundaries [80, 1, 78, 82, 2], and; *(ii)* a dedicated coordination channel must be available for the advertisement process [1] or the node rendezvous process [78].

2.3.6 Summary

Table 2.3 highlights the notable features of the protocols mentioned here. From the protocol descriptions in this section, it is evident that the bandwidth and reliability of the control node or channel can become significant obstacles to the efficacy of a multichannel protocol. This is especially so when considering decentralised processing and communications applications, which arise in many mobile and ad hoc WSN/IoT infrastructures [99, 100, 101, 102], and to a greater extent, under strong interference conditions. These issues are expected to become even more pronounced within infrastructureless deployments such as those envisaged for M2M communication and 6LoWPAN WSNs, as well as the more demanding network requirements. [103, 104, 105, 106].

Generally, data intensive wireless networks such as visual sensor networks [99, 100, 107, 108, 96], networks of mobile robots, vehicles and drones [101, 102, 97, 98, 109], and wireless capsule endoscopy [110], require:

- high bandwidth to transmit large amounts of sensory data (images, video, acceleration and position data, and so on) with low latency and the smallest possible impact on each sensor's battery resources [52, 53, 49];
- spontaneous and quick network convergence to a steady-state when multiple sensors are suddenly activated to monitor an event [111, 112, 113, 114], e.g., in vehicular networks

Table 2.3: Summary of MAC scheduling methods and notable features.

Protocol	Multiple Channel	Data Access	Coordination Method	Centralised	Timing	Bandwidth	Latency	Energy Efficiency
X-MAC	No	Contention	Preamble	Decentralised	Asynchronous	Low	Low	Low
O-MAC	No	Contention	Schedule	Decentralised	Global Clock	Med	Med	High
TRAMA	No	Reservation	Schedule	Decentralised	Global Clock	Low	Med	Low
WiseMAC	No	Contention	Preamble	Decentralised	Predictive Wakeup	Low	Med	High
ContikiMAC	No	Contention	Preamble	Decentralised	Asynchronous	Low	Low	Med
Desync	No	Reservation	Beacon	Decentralised	Local Time	Med	Low	Med
RI-MAC	No	Contention	Beacon	Decentralised	Asynchronous	Low	Med	Low
PW-MAC	No	Contention	Beacon	Decentralised	Predictive Wakeup	Low	Med	Med
EM-MAC	Yes	Contention	Beacon	Decentralised	Predictive Wakeup	Med	Med	Med
A-MAC	Yes Control	Contention	Timeslot	Centralised	Global Clock	High	Low	Low
Y-MAC	Yes Control	Contention	Timeslot	Decentralised	Global Clock	Low	Low	Med
MMAC	Yes Control	Reservation	Schedule	Decentralised	Global Clock	Med	Med	Low
CAM-MAC	Yes Control	Contention	Beacon	Decentralised	Asynchronous	High	Med	Low
MuChMAC	Yes	Reservation	Timeslot	Decentralised	Asynchronous	Med	Low	High
MMSN	Yes	Contention	Beacon	Decentralised	Global Clock	Med	Med	High
TMMAC	Yes Control	Reservation	Beacon	Decentralised	Global Clock	Med	Med	High
TSCH	Yes Control	Reservation	Schedule	Centralised	Global Clock	High	Med	High
TSMP	Yes Control	Reservation	Schedule	Centralised	Global Clock	High	Med	High

[97, 98];

- robustness to interference in the unlicensed 2.4 GHz or the 5.9 GHz dedicated short range communication (DSRC) bands [98, 79, 80], used by ad hoc wireless network deployments.

For these reasons, ongoing efforts towards a decentralised TSCH mechanism [1] that does not rely on a coordination channel (or coordinator nodes), employ distributed ALOHA-based scheduling for the advertisement channel and a gossip mechanism for the propagation and response to advertisement information. However, such mechanisms:

- are still based on time and energy consuming request and acknowledgement mechanisms
- have a rigid slotframe structure (Figure 2.6)
- require an independent manner for global time synchronisation (e.g., via a separate GPS unit [1])

Overall, with the aim to improve energy efficiency in multichannel WSNs, three key issues at the MAC layer can be identified:

- Converting the time-frequency coordination into a truly decentralised framework to avoid the dependence on a coordination channel and/or coordinator node, since these create a common point of failure and have the potential to cause bottlenecks.
- Providing a decentralised approach for time synchronisation in the network, while avoiding large network overheads for global time synchronisation packets or energy consuming hardware such as GPS receivers.
- Making node synchronisation and timeslot assignment dynamic under varying interference conditions and densities of nodes per channel to fully use the available bandwidth.

Chapter 3

Measurements and Probability Models for Energy Harvesting

3.1 Introduction

Energy harvesting is now recognised as an important aspect of WSN and IoT oriented technologies [115]. As described in Section 2.2, a multitude of research efforts have studied energy management policies [22], theoretical aspects of coupling energy production with energy consumption [49] (see Chapter 4), and practical applications [115, 36]. While most manufacturers of transducers provide specifications for the minimum, maximum and average energy harvesting characteristics of their devices (photovoltaic, piezoelectric, thermoelectric, and so on), there is still a significant gap between the reality of practical energy harvesting testbeds and the assumptions made in the research literature. For example, within the recent literature on energy harvesting based communications, there is a flurry of probability models about the harvesting process [50], but very limited experimental evidence is provided to support such models. This can be seen as a bottleneck in advancing the state-of-the-art in energy management frameworks for WSN and IoT applications, as well as limiting the applicability and impact of theoretical studies in the field.

This chapter proposes an initial coverage of this gap by providing measurements and associated software tools to capture, parse and model photovoltaic and piezoelectric energy harvesting with a real world multi-transducer platform. The focus is on the "raw" power produced by each transducer after power conditioning, as measured by high-frequency analog-to-digital (ADC) conversion that causes no interference on the actual harvesting process. The selected application environments are an outdoor and two indoor environments that represent typical office and residential conditions where IoT based applications and devices are expected to operate.

Table 3.1: Linear Technology DC2042A Energy Harvesting Multi-source Demo Board Components.

	Part Name	Purpose	IC No.
A	10V Micropower Synchronous Boost Converter	Solar Energy Harvesting	LTC3459
B	Piezoelectric Energy Harvesting Power Supply	Piezoelectric Energy Harvesting	LTC3588-1
C	Ultralow Voltage Step-Up Converter and Power Manager	Thermal Differential Energy Harvesting *(unused)*	LTC3108
D	Step-Up DC/DC Converter with Power Point Control & Low Drop Out Regulator	Misc. DC Energy Harvesting *(unused)*	LTC3105
E	Ultralow Power Supervisor with Power-Fail Output Selectable Thresholds	Supervises supply to connected device *(unused)*	LTC2935-2

The derived experimental datasets are matched with a variety of scaled probability distribution functions and results from the best fit for each case are provided. Based on the results, for all the experiments, a mixture of two to four Normal and Half-Normal distributions turns out to provide for the best fit for all cases under consideration. It is hoped that future energy management frameworks will make use of these results in order to optimise the link between energy production and consumption in IoT oriented deployments.

Section 3.2 presents the data collection process. Section 3.4 presents the results and corresponding probability models. Finally, Section 3.5 provides some concluding remarks.

3.2 Data Collection Platform & Methodology

This section provides details of the hardware and software platform used to collect empirical measurements of harvested energy available in several scenarios. Beyond the description of this chapter, the source code used for these measurements, as well as the full set of measurements, can be found on the experiment website: http://github.com/m1geo/EH_IOT.

3.2.1 Energy Harvesting

To provide the energy harvesting part of the hardware platform a Linear Technology (LT) DC2042A energy harvesting multi-source demo board was used. As detailed in Table 3.1, this board allows for energy harvesting from a variety of external transducers via a single compact circuit board, with transducers collocated in an easily accessible configuration.

Connected to this demo board are the energy harvesting transducer components, which operate as described in Table 3.2. In addition, the platform provides a light sensor to measure ambient light levels in Lux, thereby adding context to the levels of solar energy harvested. The

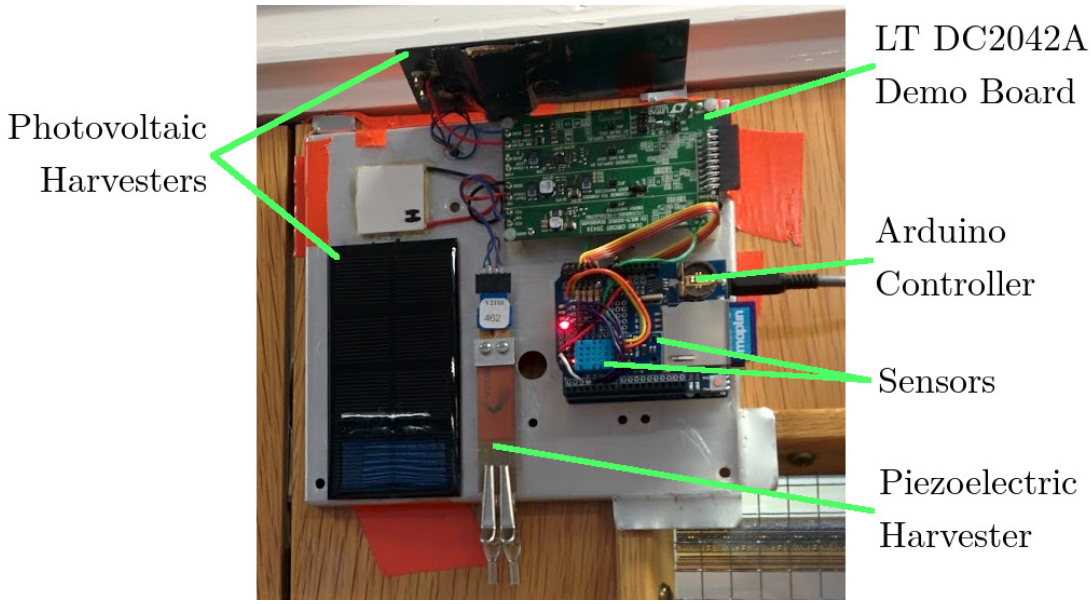


Figure 3.1: Energy harvesting platform with annotations on key components.

Table 3.2: Testbed Components.

Transducer	Description	Part
Photovoltaic	Harvests light. Two panels in series consisting of 16 x 2.5cm^2 cells. Total area 40cm^2.	Sol SM2380
Piezoelectric	Harvests vibration. Attached mass and resonant frequency varied per scenario.	Mide V21BL
Light sensor	16-bit ambient light sensor measures light fall on photovoltaic panels	ROHM BH1750

testbed and associated components are shown in Figure 3.1.

3.2.2 Portable Data Logger

The LT DC2042A harvesting board is capable of harvesting, storing and managing the power supply to low-power sensor hardware suitable for WSN and IoT oriented applications. For the measurement scenario, the non-buffered "raw" power output from each individual harvesting scheme in Table 3.2 is of interest. Since each of the power outputs could be designated to support a sensor mote, a constant load is emulated using an accurate, carefully selected resistor. The resistor causes current to flow, thus dissipating energy. By attaching the DC2042A outputs to the analogue inputs of an Arduino Uno [116] (an open source electronics prototyping platform) and using the Arduino's built-in 10-bit analogue to digital converters (ADCs), periodic samples that measure the energy dissipated into the resistor are captured for each of the individual harvesting schemes. These samples are deliberately captured without the use of any "power supervisor" ICs that would store, regulate and combine the harvested energy in order to support an attached device. This deliberate "raw" sampling allows the energy available from each

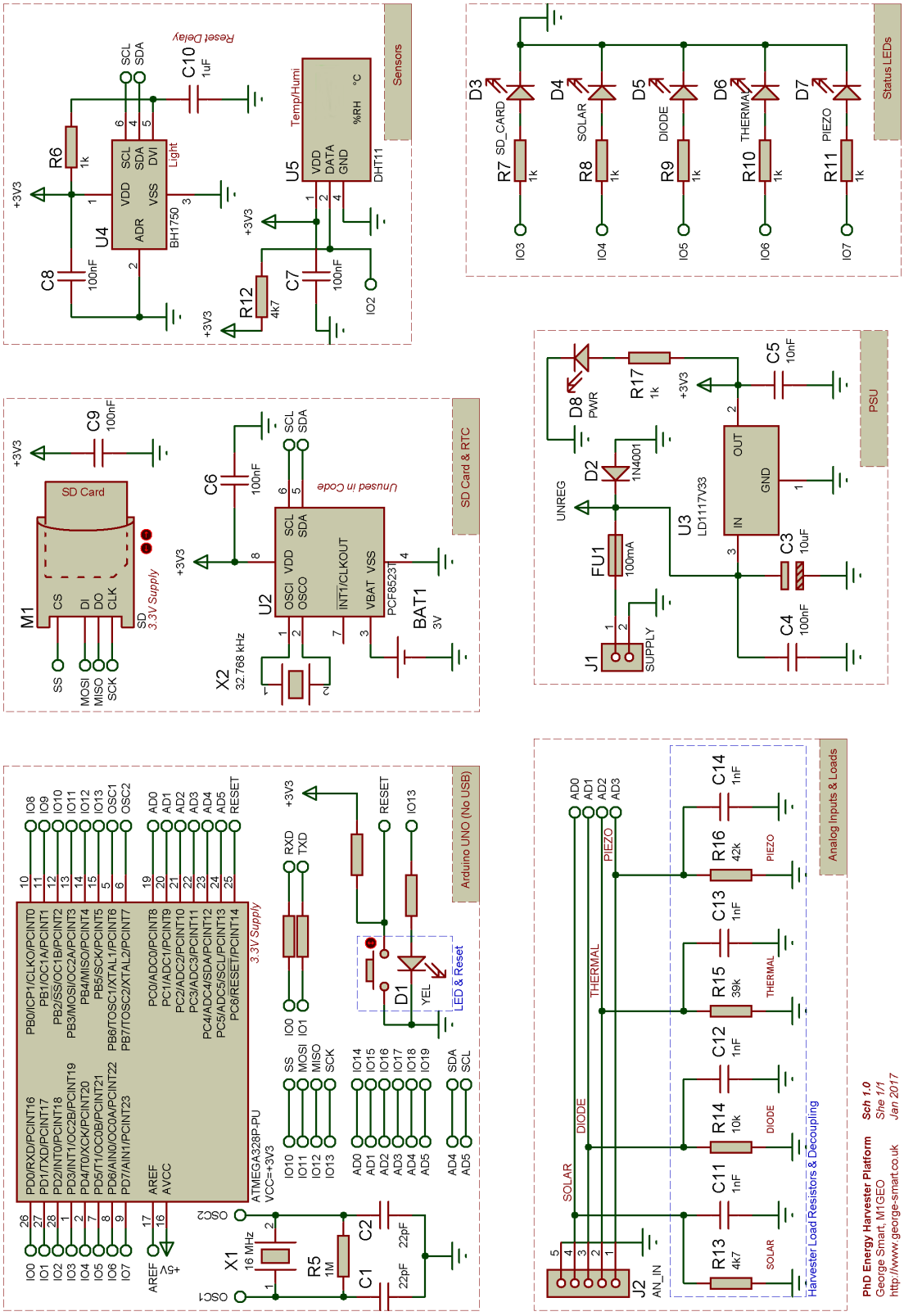


Figure 3.2: Schematic diagram of the data logger.

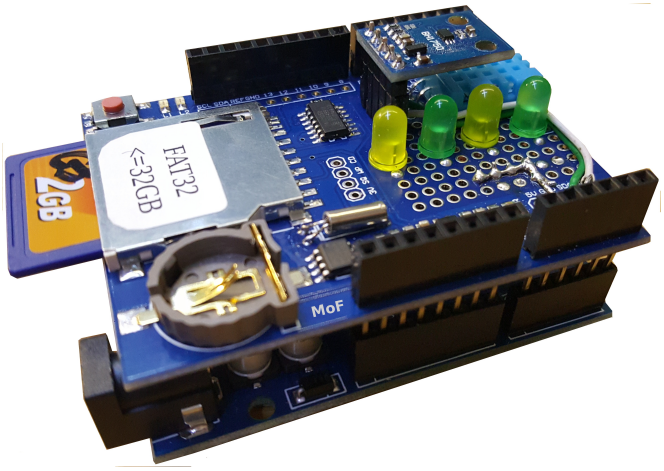


Figure 3.3: Data logger hardware comprising of Arduino Uno with SD card shield and custom electronics.

source to be recorded and analysed separately and accurately. The Arduino platform runs a custom C program that samples the energy harvested from the different schemes, as well as the light and temperature sensors, every 100 milliseconds. The samples are written to an SD card using a comma separated value (CSV) file format and a standard FAT32 file system for off-line analysis using MATLAB. Since the Arduino is used solely as a monitoring & logging device and draws power from an external power source (mains supply or external battery), it is a passive measurement device and does not affect the experiment beyond the selection of sampling frequency and ADC accuracy. Figure 3.2 shows a schematic diagram of the data logger which was built onto Arduino prototyping boards as shown in Figure 3.3. A video demonstration of the harvester platform is available on YouTube: https://www.youtube.com/watch?v=_bsCj8qFE-o. The firmware running on the Arduino Uno's ATMEGA328P MCU may be obtained from the experiment website: http://github.com/m1geo/EH_IOT, and is also included for completeness in Appendix 7.3.1.

3.3 Experimental Scenarios

This section details the environments where the data collection was performed. First, a remark that the core energy harvesting platform remains unchanged between scenarios. During outdoors experiments, the platform was enclosed in a waterproof housing that allowed for unimpeded movement of the piezoelectric harvester and that did not prevent light from falling on the photovoltaic panels or ambient light sensor. However, the piezoelectric harvester's physical resonance was tuned to a frequency appropriate to each individual scenario by adjusting the mass attached to the tip in accordance with the piezoelectric harvester datasheet [117]. Similarly, each of the load resistances were altered independently to match the energy available for

Table 3.3: Scenario Load Resistances.

Scenario	Office Door	Roof Ledge	Car
Photovoltaic	7.19 kΩ	4.67 kΩ	7.19 kΩ
Piezoelectric	42.2 kΩ	42.2 kΩ	42.2 kΩ

each scenario (see Table 3.3). These tuning techniques attempt to: *(i)* maximise the energy harvesting efficiency according to the ambient environment, and *(ii)* allow the testbed to record the best dynamic range for the harvester output.

3.3.1 Office door

In the office door scenario, the testbed was firmly affixed with metal brackets to the door of a research office containing around 15 persons. This was previously shown in Figure 3.1. The office is primarily occupied between 7am-11pm with majority of activity between 10am-8pm during weekdays and reduced activity during weekends. The office is lit by a mixture of natural sunlight during daylight hours, as well as standard office fluorescent lamps when the office is occupied. The office door automatically closes with reasonable force after every opening due to a spring-operated mechanism commonly found within public buildings for fire safety and security.

3.3.2 Roof ledge

For this scenario, the testbed was affixed securely to an outside roof ledge on an 11th floor window of the Roberts Building at University College London. The building is located in central London, with the testbed being approximately 50m above the ground. In this scenario, a simple wind-sail of approximate area 250cm^2 was attached to the piezoelectric harvester via a stiff but light-weight aluminium connecting rod. This enables us to catch gusts of wind commonly observed on rooftops, as well as low frequency vibrations caused by the turbulence of steady wind movement around the sail. For the duration of the experiment, the temperature was recorded at an average 18°C during daylight hours, with an average wind speed of 6 km/h, gusting to 25 km/h. The sky was noted as mostly clear, but with some occasional cloud cover. At the time of writing, full weather conditions for the day are available here: https://goo.gl/7huX4c.

3.3.3 Car luggage compartment

This scenario saw the testbed strapped securely into the luggage compartment of my large family car. The piezoelectric harvester was tuned to 35 Hz to match the peak of vibration

for a vehicle. The drive was largely on well-maintained tarmac roads (UK highways) at the speed limit of 70 mph (113 km/h), causing a constant vibration of around 40 Hz. Some of the journey was over rough terrain, causing a much more random frequency distribution and more intense oscillation of the piezoelectric transducer. Although the luggage compartment was not completely sealed from light, inside of the luggage compartment is covered by a parcel-shelf so minimal ambient light reached the harvester. Since the journey was made at night, the main sources of light were motorway lighting and moonlight.

3.4 Data Analysis & Visualisation

Analysis and visualisation work was performed using MATLAB. As with the Arduino program, the code to generate these model fittings and figures from the Arduino CSV data files is made available on the experiment website: http://github.com/m1geo/EH_IOT.

3.4.1 Data Preparation

Voltage sample data are imported from the Arduino via CSV files, which are loaded into MATLAB. From the voltage samples, power can calculated using Ohm's law, giving instantaneous energy dissipation readings at regular time intervals. The data is trimmed to retain only the active periods; periods of 5 minutes and longer with no harvesting are removed - examples are during dark periods for photovoltaic and periods without movement for piezoelectric. Graph fits were computed by approximating a fit manually and then exhaustively searching for the optimal fit to minimise the Kullback–Leibler (KL) divergence, D_{KL}, given by

$$D_{\mathrm{KL}} = \sum_{n} P(n) \log_2 \frac{P(n)}{Q(n)} \tag{3.1}$$

with P the theoretical probability distribution under consideration and Q the experimentally-measured and normalised histogram of energy values sampled at points n.

3.4.2 Empirical Observations and Models under Consideration

Table 3.4 shows mean and maximum energy values obtained for each of the three scenarios under consideration. Evidently, the three scenarios under consideration represent different cases for each modality of energy harvesting. For example, the "car luggage compartment" scenario represents the low-end of the harvesting spectrum, where both photovoltaic and piezoelectric power is modest. The indoor "office door" scenario represents the mid-range scenario where medium photovoltaic and moderate piezoelectric harvesting is achievable. Finally, the outdoor

Table 3.4: Empirical Scenario Conditions. Values reported are average, with maxima in brackets. All minima are zero.

Scenario	Ambient Light (Lux)	Photovoltaic Power (μW)	Piezoelectric Power (μW)
Office door	56.47 (231)	41.15 (418.7131)	2.43 (112.6020)
Roof ledge	5697.10 (54612)	953.58 (2422.857)	6.38 (133.1557)
Car	1.30 (370)	7.97 (1563.537)	5.32 (156.6202)

"roof ledge" scenario represents the most volatile case where, on average, high photovoltaic and piezoelectric powers can be harvested.

In terms of modelling, mixture models of several distributions, including Exponential, inverse-Gamma, Normal, Half-Normal, Poisson and Pareto were considered. Out of a multitude of fitting experiments via the minimisation of (3.1), mixtures using the following two distributions were found to provide for the best results:

- the Normal distribution with mean μ and standard deviation σ:

$$P_{\mathrm{N}}(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{3.2}$$

- the Half-Normal distribution with mean $\sigma\sqrt{\frac{2}{\pi}}$ and standard deviation $\sigma\sqrt{1-\frac{2}{\pi}}$:

$$P_{\mathrm{HN}}(\sigma) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} e^{\frac{-x^2}{2\sigma^2}} \tag{3.3}$$

For each harvesting scenario, the associated scaling parameter, s, required to normalise the experimentally measured data to the theoretical probability distributions is provided.

3.4.2.1 Office door

Photovoltaic For the case of photovoltaic harvesting, the best fit for the "office door" experimental data was obtained with a mixture of three Normal distributions. The resulting fit is shown in Figure 3.4 and it corresponds to

$$\begin{aligned} P_{\mathrm{door,PV}} &= s_{\mathrm{door,PV}} \left[a_1 P_{\mathrm{N}}(\mu_1,\sigma_1)\right. \\ &+ \left. a_2 P_{\mathrm{N}}(\mu_2,\sigma_2) + a_3 P_{\mathrm{N}}(\mu_3,\sigma_3)\right] \end{aligned} \tag{3.4}$$

with the parameters given in Table 3.5 and scaling factor $s_{\mathrm{door,PV}} = 6.076 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\mathrm{KL}} = 4.823 \times 10^{-2}$.

Piezoelectric For the case of piezoelectric harvesting, the best fit was obtained with a mixture of two Half-Normal and one Normal distribution. The resulting fit is shown in Figure 3.5 and

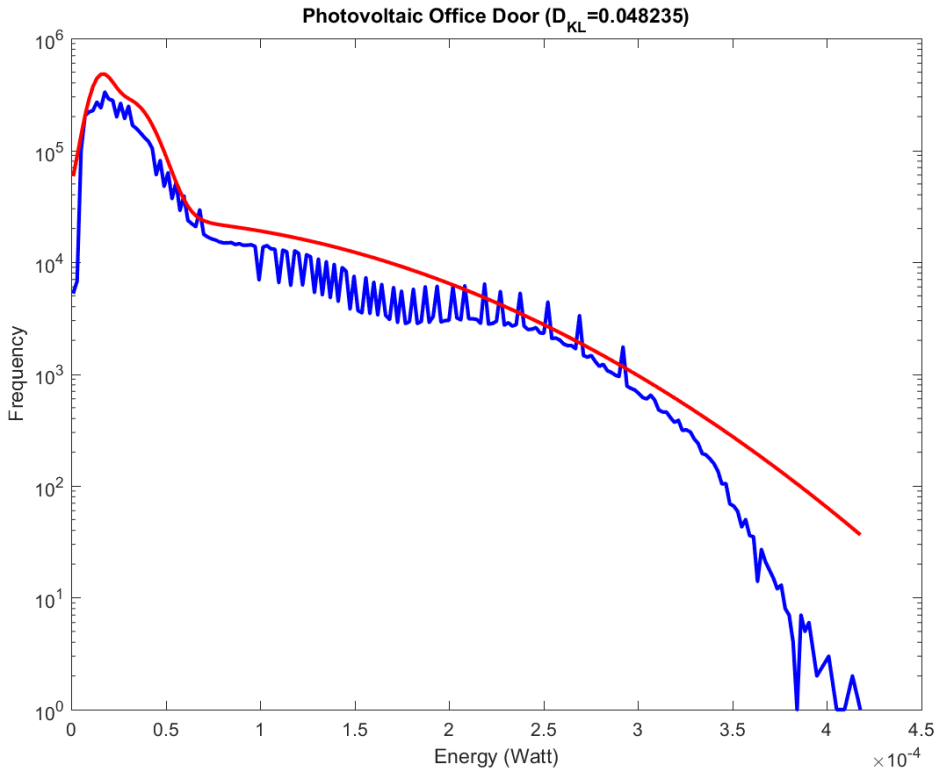


Figure 3.4: Histogram of photovoltaic harvester (blue) on the "office door" and best fit (red) obtained via (3.4).

Table 3.5: Photovoltaic harvester on the office door.

i	a_i	μ_i	σ_i
1	5.038	1.541e-05	6.059e-06
2	7.582	3.022e-05	1.213e-05
3	6.943	1.779e-05	1.107e-04

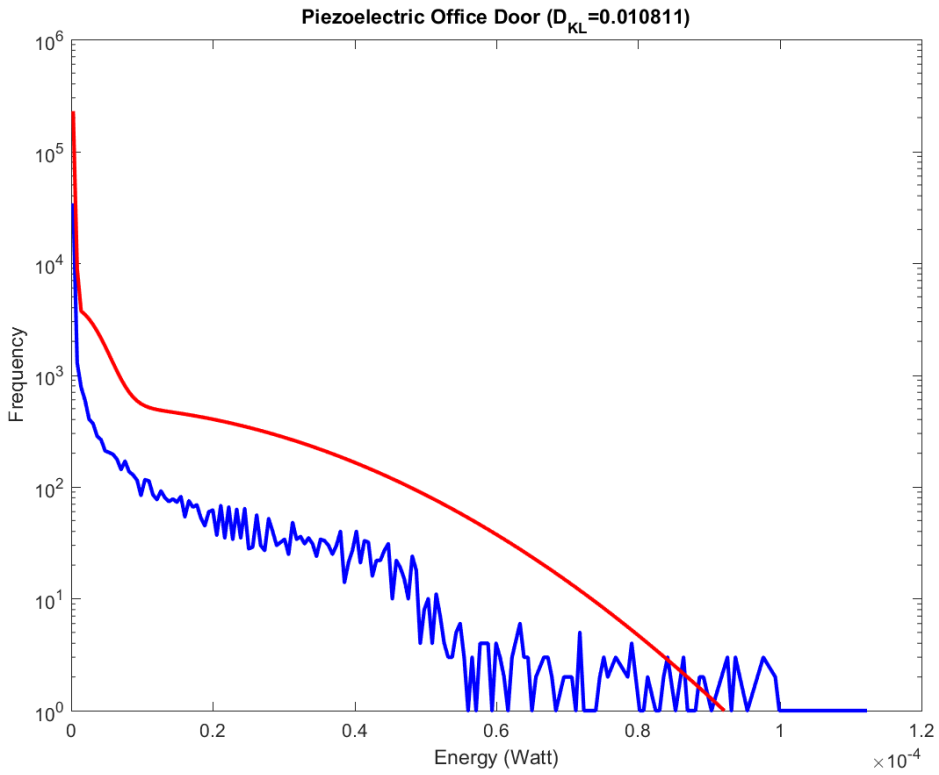


Figure 3.5: Histogram of piezoelectric harvester (blue) on the "office door" and best fit (red) obtained via (3.5).

Table 3.6: Piezoelectric harvester on the office door.

i	a_i	μ_i	σ_i
1	1.306e-01	0	2.894e-07
2	1.471e-02	0	3.384e-06
3	3.522e-02	4.867e-09	2.598e-05

it corresponds to

$$\begin{aligned} P_{\text{door,PE}} &= s_{\text{door,PE}}\,[a_1 P_{\text{HN}}(\sigma_1) \\ &+ a_2 P_{\text{HN}}(\sigma_2) + a_3 P_{\text{N}}(\mu_3, \sigma_3)] \end{aligned} \tag{3.5}$$

with the parameters given in Table 3.6 and scaling factor $s_{\text{door,PE}} = 613.787 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\text{KL}} = 1.081 \times 10^{-2}$.

3.4.2.2 Roof ledge

Photovoltaic For the case of photovoltaic harvesting, the best fit for the "roof ledge" experimental data was obtained with a mixture of three Normal distributions and one Half-Normal

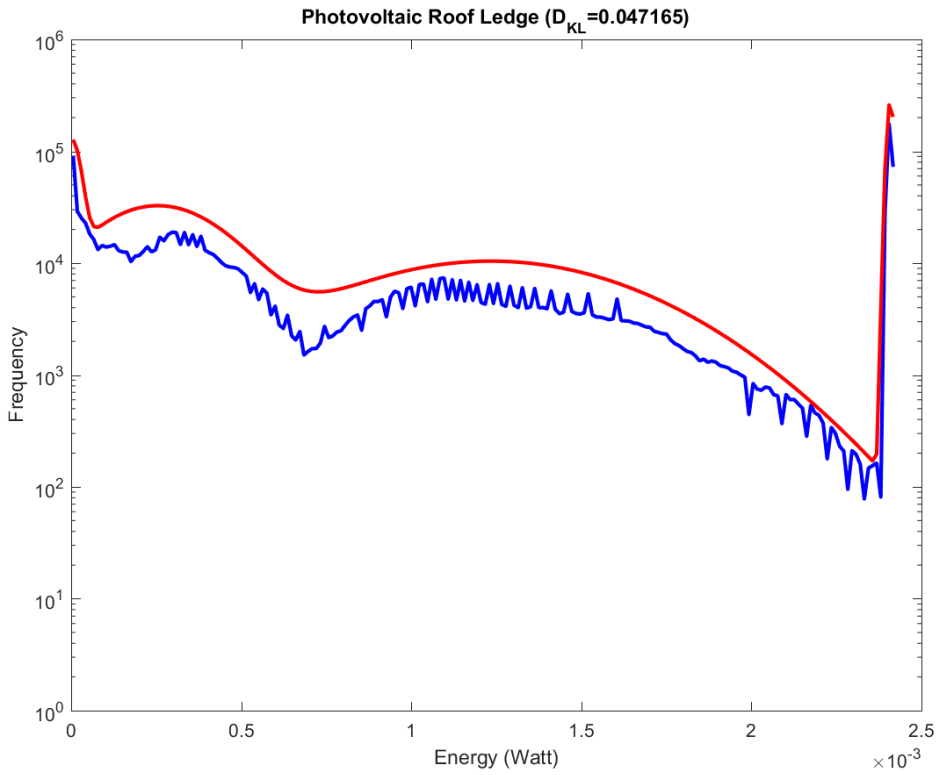


Figure 3.6: Histogram of photovoltaic harvester (blue) on "roof ledge" and best fit (red) obtained via (3.6).

Table 3.7: Photovoltaic harvester on roof ledge.

i	a_i	μ_i	σ_i
1	3.444	0	2.327e-05
2	14.626	2.517e-04	1.812e-04
3	10.230	1.234e-03	3.914e-04
4	6.888	2.409e-03	9.627e-06

distribution. The resulting fit is shown in Figure 3.6 and it corresponds to

$$\begin{aligned} P_{\text{roof,PV}} &= s_{\text{roof,PV}} \left[a_1 P_{\text{HN}}(\sigma_1) + a_2 P_{\text{N}}(\mu_2, \sigma_2) \right. \\ &+ \left. a_3 P_{\text{N}}(\mu_3, \sigma_3) + a_4 P_{\text{N}}(\mu_4, \sigma_4) \right] \end{aligned} \tag{3.6}$$

with the parameters given in Table 3.7 and scaling factor $s_{\text{roof,PV}} = 2.943 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\text{KL}} = 4.716 \times 10^{-2}$.

Piezoelectric For the case of piezoelectric harvesting, the best fit was obtained with a mixture of one Half-Normal and two Normal distributions. The resulting fit is shown in Figure 3.7 and

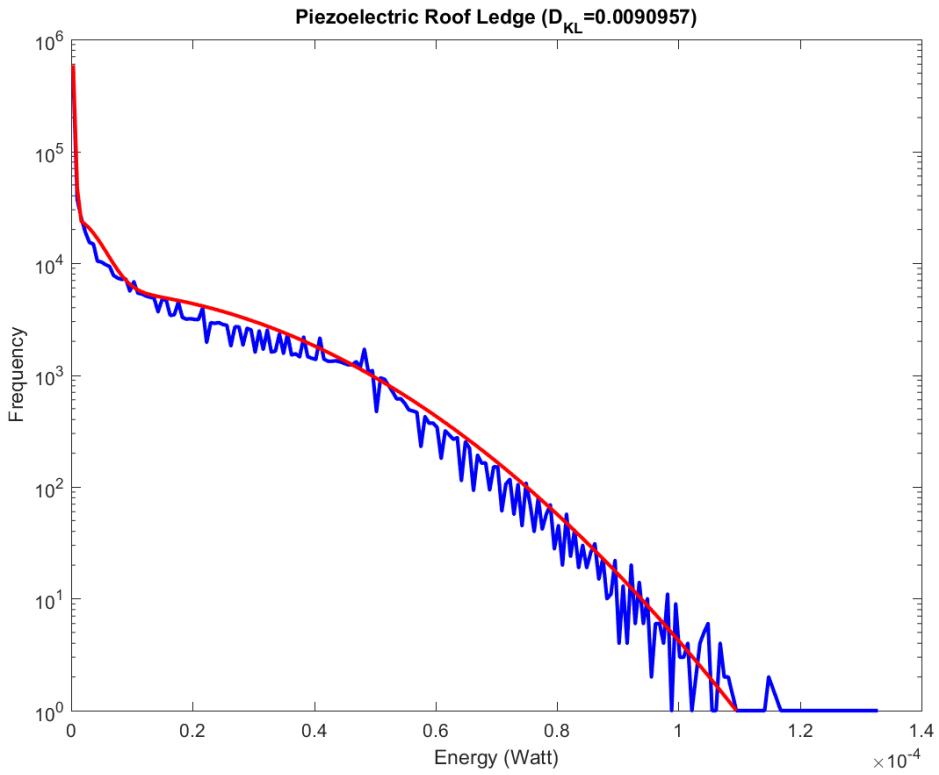


Figure 3.7: Histogram of piezoelectric harvester (blue) on "roof ledge" and best fit (red) obtained via (3.7).

Table 3.8: Piezoelectric harvester on roof ledge.

i	a_i	μ_i	σ_i
1	3.915e-01	0	3.7025e-07
2	9.750e-02	9.971e-09	4.002e-06
3	1.915e-01	4.867e-09	2.631e-05

it corresponds to

$$\begin{aligned} P_{\text{roof,PE}} &= s_{\text{roof,PE}}\,[a_1 P_{\text{HN}}(\sigma_1) \\ &+ a_2 P_{\text{N}}(\mu_2,\sigma_2) + a_3 P_{\text{N}}(\mu_3,\sigma_3)] \end{aligned} \tag{3.7}$$

with the parameters given in Table 3.8 and scaling factor $s_{\text{roof,PE}} = 186.529 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\text{KL}} = 0.910 \times 10^{-2}$.

3.4.2.3 Car luggage compartment

Photovoltaic For the case of photovoltaic harvesting, the best fit for the "car luggage compartment" experimental data was obtained with a mixture of two Normal distributions and one

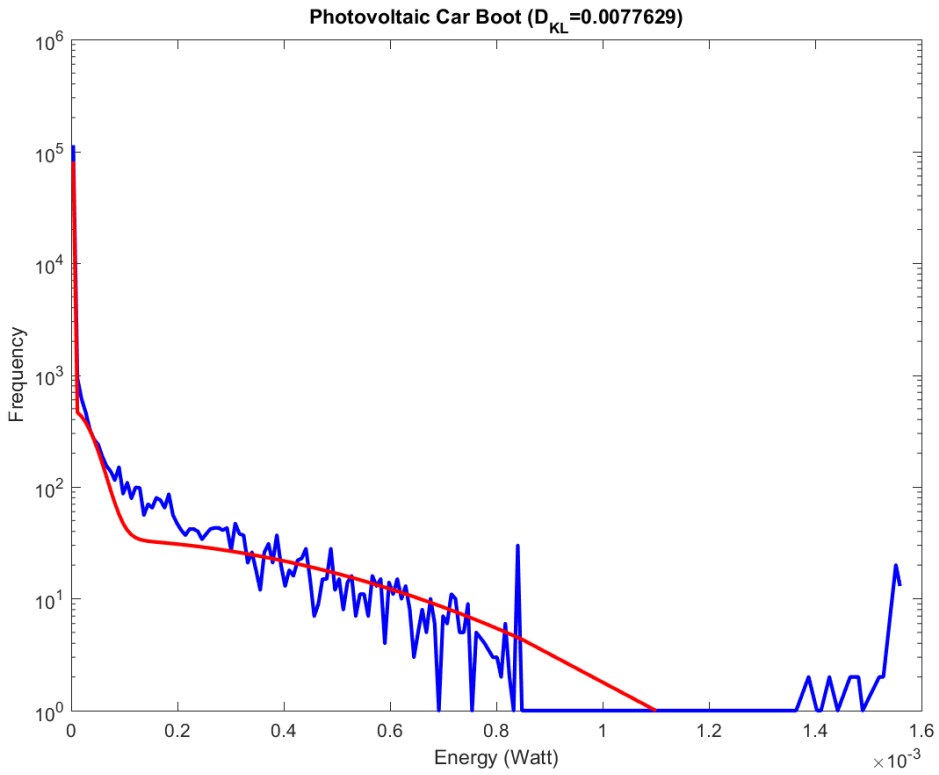


Figure 3.8: Histogram of photovoltaic harvester (blue) in "car luggage compartment" and best fit (red) obtained via (3.8).

Table 3.9: Photovoltaic harvester in car luggage compartment.

i	a_i	μ_i	σ_i
1	1.363	0	2.000e-06
2	4.170e-02	4.578e-09	3.691e-05
3	3.607e-02	1.075e-08	4.163e-04

Half-Normal distribution. The resulting fit is shown in Figure 3.8 and it corresponds to

$$\begin{aligned} P_{\text{car,PV}} &= s_{\text{car,PV}}\left[a_1 P_{\text{HN}}(\sigma_1)\right. \\ &+ \left. a_2 P_{\text{N}}(\mu_2,\sigma_2) + a_3 P_{\text{N}}(\mu_3,\sigma_3)\right] \end{aligned} \tag{3.8}$$

with the parameters given in Table 3.9 and scaling factor $s_{\text{car,PV}} = 71.332 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\text{KL}} = 0.777 \times 10^{-2}$.

Piezoelectric Finally, for the case of piezoelectric harvesting, the best fit was obtained with a mixture of one Normal and two Half-Normal distributions. The resulting fit is shown in Figure

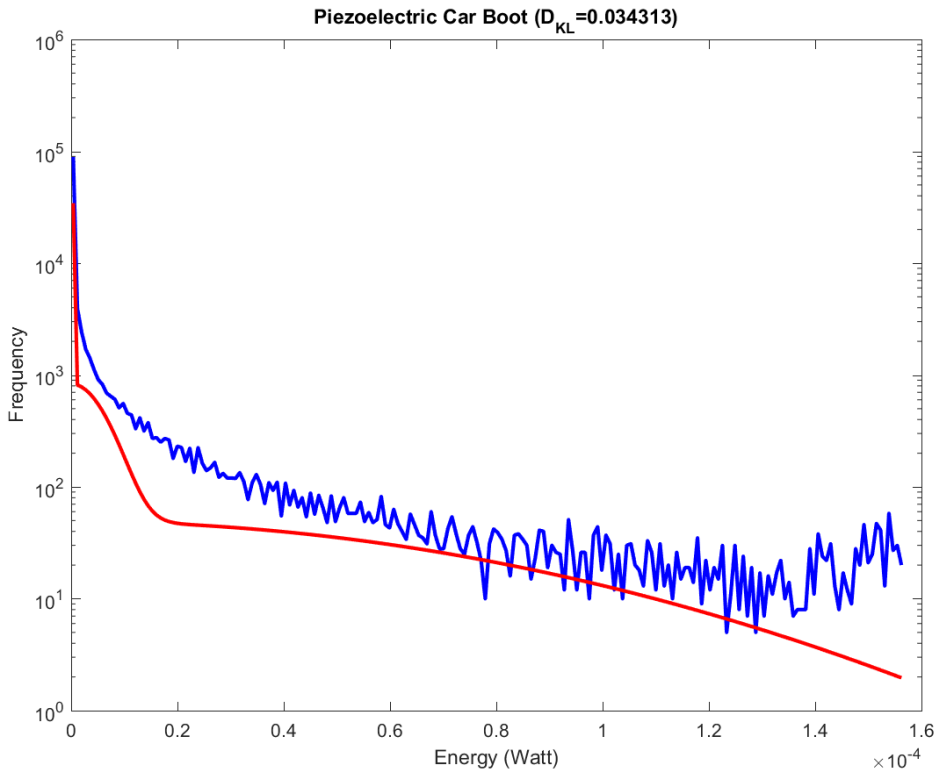


Figure 3.9: Histogram of piezoelectric harvester (blue) on "car luggage compartment" and best fit (red) obtained via (3.9).

Table 3.10: Piezoelectric harvester in car luggage compartment.

i	a_i	μ_i	σ_i
1	5.715e-02	0	2.000e-07
2	5.261e-03	0	5.351e-06
3	3.784e-03	4.867e-09	6.159e-05

3.9 and it corresponds to

$$\begin{aligned} P_{\text{car,PE}} &= s_{\text{car,PE}}\,[a_1 P_{\text{HN}}(\sigma_1) \\ &+ a_2 P_{\text{HN}}(\sigma_2) + a_3 P_{\text{N}}(\mu_3, \sigma_3)] \end{aligned} \tag{3.9}$$

with the parameters given in Table 3.10 and scaling factor $s_{\text{car,PE}} = 1555.130 \times 10^{-2}$. The KL divergence for this case was found to be: $D_{\text{KL}} = 3.431 \times 10^{-2}$.

3.5 Conclusion

This work was motivated by the lack of experimental evidence on the capabilities of practical transducer technologies in scenarios appropriate to IoT deployments. To complement this current gap of data and associated probability models, a multi-transducer platform was deployed,

equipped for photovoltaic and piezoelectric energy harvesters – technologies that are expected to be deployed within IoT data gathering and transmission frameworks. The provided experiments and the associated online repository at http://github.com/m1geo/EH_IOT include a full dataset that can be used for research in energy-neutral operation of WSN and IoT platforms, as well as feasibility studies in energy optimisation of practical deployments, before engaging in cumbersome deployments in the field. Source code for the harvester testbed is included for completeness in Appendix 7.3.1.

Chapter 4

Analytic Conditions for Energy Neutrality in Uniformly-formed WSNs

4.1 Introduction

In this chapter, the problem of energy neutrality is approached in a holistic, system-oriented, manner. Specifically, focus is on the common application scenario of a monitoring infrastructure where sensor nodes follow a periodic duty cycle in order to capture and transmit measurements to a base station, or to another node that relays the information to a base station. A parametric model for energy neutrality is derived in function of the system settings under the assumption of a uniformly-formed WSN, i.e., a network of identical sensor nodes that are: *(i)* producing data traffic with the same statistical characterisation and *(ii)* connected to the base station via a cluster-tree topology [62] represented by a symmetric and acyclic graph with balanced bandwidth allocation per link. Within this framework, the key advance of this chapter's work in comparison to previous works on optimal energy management policies [22, 51, 66, 64, 65] is that it provides closed-form expressions for the minimum-required harvested energy in order for each node to remain energy neutral.

Section 4.2 presents the system model corresponding to the application scenarios under consideration. The analytic derivations characterising energy neutral operation under different data transmission rates are presented in Section 4.3, where the minimum requirement for harvested energy under various widely used statistical characterisations for the data transmission rate is also derived. Section 4.4 presents the experimental validation of the proposed analytic formula-

tions for energy-neutral operation based on TelosB testbed measurements, Section 4.5 presents results within two applications. Finally, Section 4.6 provides some concluding comments from the chapter.

4.2 System Model

This chapter considers a set of wireless sensor nodes connected to a "sink" node, which represents the collecting unit, i.e., a base station with power supply. This connection could be direct; alternatively, under a symmetric and balanced cluster-tree topology [21, 118], each node could be linked to a "relay" node that conveys measurements (along with its own) to another relay node or, eventually, to a base station. Interference between neighbouring nodes can be avoided by using simple heuristics or graph colouring approaches in conjunction with transmission and reception in different channels. For example, a node can listen to Channel X and transmit in Channel Y, with $X \neq Y$ [31, 92, 93, 119, 120, 121, 122, 123]. Such examples are illustrated in Figure 4.1. The ellipses indicate the coverage of each receiver, with their channel allocated such that no inter-cluster interference is possible. The links indicate the bandwidth available to each transmitting sensor node. The figure shows the essentials of the problem can be reduced to the analysis of the interaction between each sensor node and its corresponding base station or relay node at the same tier of the cluster-tree topology.

4.2.1 System Description

In the analysis it is assumed that, for a harvesting interval of T seconds, the sensor nodes are continuously active for T_{act} seconds. This defines the duty cycle

$$c = \frac{T_{\text{act}}}{T}. \tag{4.1}$$

This activation can be triggered by external events or by scheduled data gathering with rate c over the duration of the application, $0 < c < 1$. Examples are: data acquisition and transmission in environmental monitoring [21], event driven activation for surveillance [22], and adaptive control of duty cycling for energy management [66, 124]. Thus, the value of c can be adjusted statically or dynamically based on empirical observations from the application environment.

When the sensor nodes are activated, they first converge into a balanced time-frequency steady-state mode, where each node joins one base station (or a relay node) on a particular channel such that: *(i)* the number of nodes coupled to each base station or relay node is balanced, and *(ii)* each cluster-tree tier accommodates transmissions from n nodes without collisions. Several low-energy (centralised or distributed) WSN protocols, such as EM-MAC

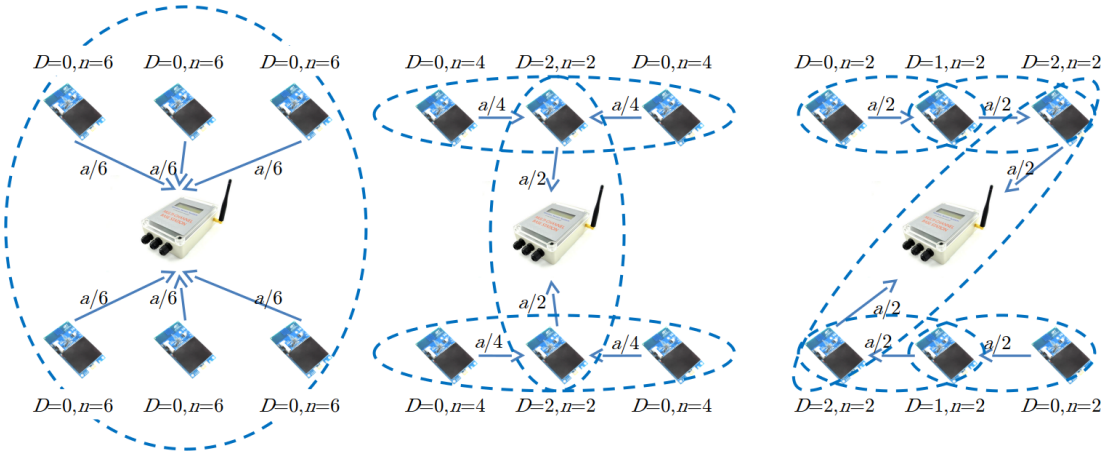


Figure 4.1: Examples of several uniformly-formed topologies that can operate in collision-free steady-state mode.

[3], wirelessHART [93], IEEE 802.15.4 GTS [62], TFDMA [31] and the proposed DT-SCS protocol described in Chapter 5 can achieve this goal. For example, TFDMA achieves this for 16 nodes and 4 channels within 3-5 seconds [31], centralised IEEE 802.15.4 GTS can establish collision-free single-channel time division multiple access (TDMA) within 1-2 seconds [62] and DT-SCS within 1-2 seconds (see Figure 5.6). While energy is consumed for this convergence, the payoff for the WSN is the achievement of balanced, collision-free, steady-state operation with predictable characteristics during the active period. Figure 4.1 shows three interference-free uniformly-formed topologies within a network comprising six identical sensor nodes and one base station, with a indicating the consumption rate of each sink node (in bits-per-second). The left topology shows a direct (one-tier) connection to the base station; the centre topology shows a two-tier cluster-tree topology, and; the right topology shows a three-tier cluster-tree topology. Parameter d indicates the additional nodes whose traffic is relayed by each node, as well as the number of nodes sharing the same sink node (receiver) at the same tier of the cluster-tree topology (via n).

Each sensor captures, processes and transmits (and potentially relays) data. It is assumed that the transmission data rate varies, and is thus modelled as a random variable. The rate variability may stem from: adaptive sensing strategies [125], packet retransmissions or protocol adaptivity to mitigate interference effects [3], and variable-rate source-channel encoding [126] to reduce the transmission bitrate and ensure robustness to packet erasures [127]. Thus, due to these factors, the number of bits sent within each transmission slot of the utilised protocol varies, despite the fact that the physical later rate is fixed for most WSN systems using the IEEE 802.15.4 PHY [24].

Within each tier of the cluster-tree topology, depending on the amount of data to be transmitted, a node may need to: *(i)* stay awake transmitting beacon packets (with radio on) if less

bits have to be sent than what is possible within its transmission slot, or *(ii)* buffer the residual data if more bits must be sent than what its slot permits. Once the active period of T_{act} seconds lapses, each node suspends its activity (i.e., goes into "sleep" mode) in order to conserve energy. Figure 4.2 shows two examples of TDMA transmission slots during the active period. During both the active and sleep modes, each sensor harvests energy based on its onboard harvesting unit (e.g., piezoelectric harvester or photovoltaic panels), as in Chapter 3.

Each sensor (e.g., TelosB, micaZ, STM32W motes, and so on) can be powered by its onboard batteries for long time intervals (e.g., hundreds of hours of continuous operation). As such, we can assume the battery capacity to be infinite when compared to the energy budget spent and harvested within each interval of T seconds [22, 74, 67]. In addition, due to the assumption of infinite battery capacity, issues such as leakage current and battery ageing do not need to be considered.

Practical WSN and IoT transceiver hardware reacts in intervals proportional to one packet transmission (or to the utilised time-frequency slotting mechanism). Thus, the transmission and reception of data is not strictly a continuous process. However, energy consumption within each sensor node is strictly continuous as, regardless of the transceiver, each sensor node is active for the entire duration of T_{act} seconds by sensing, processing data (e.g., to remove noise or to perform data encoding) and other runtime operations related to data gathering, processing and transmission (such as buffer management at the application, medium access and physical layers and servicing interrupts in the runtime environment).

4.2.2 Definitions

When the WSN goes into the active state, it is assumed that k Joule is consumed by each sensor node in order to reach the balanced, collision-free, steady-state operation via one of the well-known centralised or distributed mechanisms suitable for this purpose [31, 92, 93, 62, 3]. During the steady-state operation of each node, the average energy rate consumed to process and transmit data is g Joule-per-bit.

4.2.2.1 Data Production and Energy Harvesting

Since the data production and transmission by each sensor node is a nondeterministic process (i.e., their behaviour is not consistent across multiple runs), the data transmission rate (in bits-per-second) is modelled by random variable (RV) Ψ with PDF $P(\psi)$. The statistical modelling of this rate can be gained by observing the occurred physical/chemical phenomena and analysing the behaviour of each node when it captures, processes and transmits bits relevant to them, in conjunction with the data relayed by other nodes of the same tier (if the node is also a receiver).

Alternatively, the data production and transmission rate can be controlled (or "shaped") by the system designer in order to achieve a certain goal, such as limiting the occurring latency or, in the case of these experiments, to minimise the harvested energy required in order to operate each node in perpetuity. Examples of systems with variable data transmission rates include visual sensor networks transmitting compressed video frames or image features [128, 129, 130, 131], as well as an activity monitoring or localisation networks where the data acquisition is irregular and depends on the events occurring in the monitored area [132, 133, 134].

The energy harvesting process is also a nondeterministic process [39]. For example, as shown in Chapter 3, solar or vibration energy scavenging mechanisms produce different levels of power at different times of the day, depending on the environmental conditions and on whether they are placed indoors or outdoors [39, 37]. Therefore, the power (Watt) produced by the harvesting mechanism is modelled by RV X with PDF $P(\chi)$.

Since both the data rate and the power produced by the harvester may be nonstationary (i.e., their PDFs change with time), marginal statistics for $P(\psi)$ and $P(\chi)$ are assumed, which are derived starting from a doubly stochastic model for these processes. Specifically, such marginal statistics can be obtained by [135, 136]: *(i)* fitting PDFs to sets of past measurements of data rates and power (such as those presented in Chapter 3), with the statistical moments (parameters) of such distributions characterised by another PDF; *(ii)* integrating over the parameter space to derive the final form of $P(\psi)$ and $P(\chi)$. For example, if the data transmission rate is modelled as a Half-Gaussian distribution with variance parameter that is itself exponentially distributed, by integrating over the parameter space, the marginal statistics of the data rate become Laplacian [135, 136]. The disadvantage of using marginal statistics for the data transmission rate and the power produced by the harvester is the removal of the stochastic dependencies to transient physical properties of these quantities. However, this work is interested in the expected requirements for energy harvesting to maintain energy neutrality over a lengthy time interval (e.g., several hours) and not in the variations of energy harvesting over short time intervals. Such variations are irrelevant since the on-board sensor batteries can support stand-alone node operation for hundreds of hours, if needed. Thus, a mean-based analysis using marginal statistics is suitable for this purpose.

4.2.2.2 Data Consumption and Energy Penalties

The data consumption rate of the application layer of each receiver under the employed collision-free steady-state operation is a bits-per-second (bps). For example, under the IEEE 802.15.4 PHY [24] and the CC2420 transceiver [29], $a \cong 144$ kbps at the application layer under the

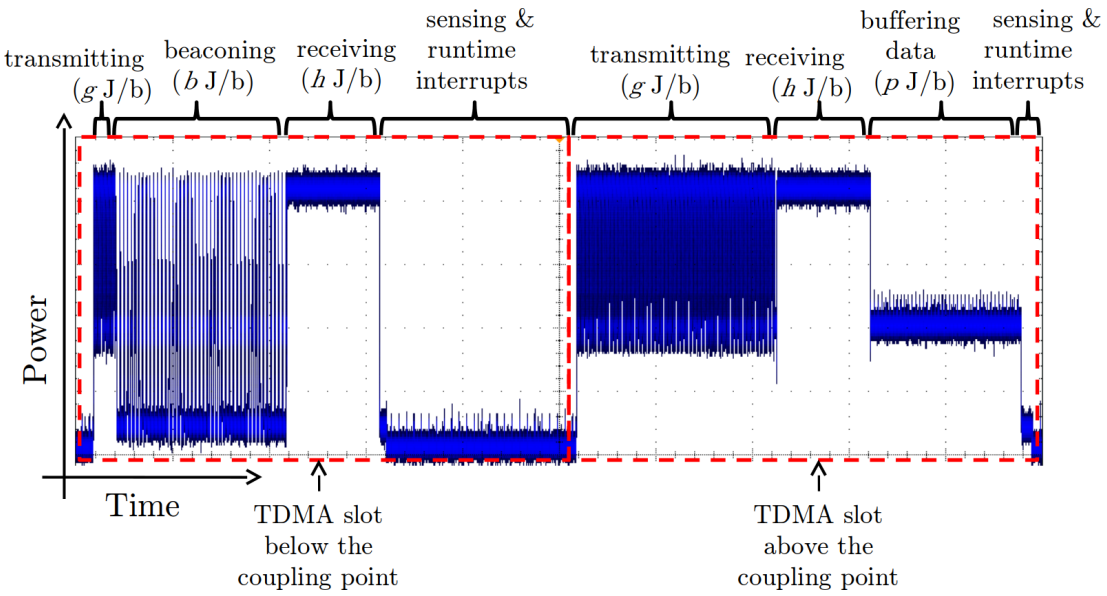


Figure 4.2: Energy profile of a TelosB sensor node within an undercoupled and an overcoupled TDMA slot during the active period. The indicated metrics (in Joule-per-bit) are defined in Table 4.1.

NullMAC and NullRDC options of Contiki-OS IoT operating system[1]. Since each sink node is coupled with n identical sensor nodes at the same tier of the cluster-tree topology (Figure 4.1), the coupling point of each sink (receiver) node is defined as the ratio $\frac{a}{n}$. This means that, in the ideal case, each sensor node should transmit its captured data at the rate of $\frac{a}{n}$ bps. However, given the time-varying nature of the data transmission rate per node, beyond the energy for data capturing and transmission there exists the following two cases: *(i)* sink underloading, where $\Psi < \frac{a}{n}$ and "idle" energy is consumed by the node with rate b Joule-per-bit (J/b) by staying active during transmission opportunities for synchronisation and other runtime purposes (e.g., transmitting beacon messages [118, 31]); *(ii)* sink overloading, where $\boldsymbol{\Psi} > \frac{a}{n}$ and "penalty" energy is consumed with rate p J/b by the sensor to buffer (and retrieve) the data prior to transmission. Examples of both are illustrated in Figure 4.2 for TDMA-based collision-free transmission [31, 32]. The nomenclature summary of the system model is given in Table 4.1.

4.3 Characterisation of Energy Neutrality

From the previously described system model, the analytic conditions required to maintain energy neutrality are derived corresponding to the minimum energy harvesting requirement. There are two modes of operation with complementary energy profiles: the active mode, where

[1] https://goo.gl/GMfEAE contains more details: the NullMAC removes any MAC-level processing, while the NullRDC turns off the radio duty cycling. This leads to the maximum energy efficiency, assuming that the application layer handles the transmission opportunities and transceiver state appropriately.

Table 4.1: Nomenclature table.

Symbol	Unit	Definition
c	–	Duty cycle
T, T_{act}	s	Harvesting time interval, active time interval. Harvesting time varies by transducer technology (e.g., 24 hours for solar, piezoelectric machine vibration varies with machine use).
n	–	Number of sensor nodes at the same tier of the cluster-tree topology
d	–	Number of additional sensor nodes whose traffic is relayed by each node at a given tier of the cluster-tree topology
k	J	Energy consumed for wakeup, setup and convergence
g	J/b	Energy for processing and transmitting one bit
p	J/b	Penalty energy for storing one bit during sink overloading
b	J/b	Energy during idle periods for the time interval corresponding to one bit transmission
h	J/b	Energy for receiving and temporary buffering one bit under the relay case
a	bps	Data consumption rate of a relay node (or base station)
r	bps	Average data transmission rate per node
$\Psi \sim P_{d+1}(\psi)$	bps	RV modelling the data production and transmission rate per node that is also relaying data from d other nodes
$E_{d+1}[\Psi]$	bps	Expected data production and transmission rate per node that is also relaying data from d other nodes
$X \sim P(\chi)$	W	RV modelling the power harvested by each node
$E[X]$	W	Expected power harvested by each node, as presented in Chapter 3
E_n	J	Node residual energy (harvested minus consumed) over the harvesting time interval T

energy is (primarily) consumed, and the sleep (or suspend) mode, where each node is suspended and energy is harvested in order to replenish the node's battery resources. During both the sleep and the active modes, each sensor node is expected to harvest $T\int_0^\infty \chi P(\chi)\,d\chi = TE[X]$ Joule from the surrounding environment.

During the active mode period of cT seconds, five components for the energy consumption for each sensor node are defined, most of which are pictorially illustrated in Figure 4.2:

1. *Setup and convergence energy* — consumed when each node is activated (once during the harvesting time interval). The energy to converge to steady-state is k J. The convergence time is at least two orders of magnitude smaller than T_{act} (e.g., 1-5 s vs. $T_{\text{act}} = 400$ s) and can be considered negligible in comparison to T_{act}.

2. *Energy for processing and transmitting* — consumed when processing and transmitting the node's own data and the data relayed to it from d other nodes, given by $cTg\int_0^\infty \psi P_{d+1}(\psi)\,d\psi = cTgE_{d+1}[\Psi]$ J, with $E_{d+1}[\Psi] \equiv (d+1)\,E[\Psi]$ and $E[\Psi]$ the expected transmission rate of each node that is not a relay. If $E_{d+1}[\Psi] > \frac{a}{n}$ (i.e., the mean transmission rate is higher than the coupling point), then T_{act} includes the time each node has to remain active without producing new data, in order to complete the transmission of the data buffered in its flash memory.

3. *Energy for receiving and buffering data* — consumed when receiving and buffering data in low-power on-chip memory from d nodes prior to relaying it, given by $cTh\int_0^\infty \psi P_d(\psi)\,d\psi = cThE_d[\Psi]$ J. This energy is dominated by the receiver power requirements. Moreover, in practical WPAN and WLAN transceiver hardware, the average transceiver power in receive mode is virtually the same whether the node is actually receiving data or merely hearing noise. It is thus irrelevant to the receiver power whether the transmitting node used its entire transmission slot or not.

4. *Idle energy* — consumed when the data rate Ψ is smaller than the sink coupling point $\frac{a}{n}$: $cTb\int_0^{\frac{a}{n}} \left(\frac{a}{n} - \psi\right) P_{d+1}(\psi)\,d\psi$ J. This energy corresponds to beaconing to maintain network synchronisation and other runtime operations carried out during the transmit mode.

5. *Penalty energy* — consumed when the data rate Ψ is larger than the sink coupling point $\frac{a}{n}$ and the data is buffered in high-power, typically off-chip, memory prior to transmission at the next available opportunity: $cTp\int_{\frac{a}{n}}^\infty \left(\psi - \frac{a}{n}\right) P_{d+1}(\psi)\,d\psi$ J.

Notice that, apart from the setup and convergence energy, the energy consumption for all the remaining components is affected by the total number of additional nodes (d) relaying their traffic via a given node. Example cluster-tree topologies providing instantiations for d and n in WSNs are given in Figure 4.1 and discussed in Section 4.2.

The residual energy of each node, E_n, in a tier of the cluster-tree topology is defined as the difference between the produced (harvested) energy and the consumed energy over the harvesting time interval. It can be calculated for each sensor node by:

$$\begin{aligned} E_n &= TE\left[\chi\right] - k - cT \times \left[E_{d+1}\left[\Psi\right]\left(g + \frac{hd}{d+1}\right)\right. \\ &+ b\int_0^{\frac{a}{n}} \left(\frac{a}{n} - \psi\right) P_{d+1}\left(\psi\right) d\psi \\ &+ \left. p\int_{\frac{a}{n}}^{\infty} \left(\psi - \frac{a}{n}\right) P_{d+1}\left(\psi\right) d\psi\right]. \end{aligned} \tag{4.2}$$

Clearly, $E_n < 0$ corresponds to energy deficit (the expected energy produced by the harvesting process is less than the expected consumption during the harvesting time interval), $E_n > 0$ corresponds to energy surplus and $E_n = 0$ corresponds to energy neutrality. Notice that the relationship $\forall d > 0 :\ E_{d+1}\left[\Psi\right] = \frac{d+1}{d} E_d\left[\Psi\right]$ is used in (4.2), since the expected transmission rate of each node increases linearly with respect to d in a uniformly-formed WSN. Adding and subtracting $cTp\int_0^{\frac{a}{n}} \left(\psi - \frac{a}{n}\right) P_{d+1}\left(\psi\right) d\psi$ in E_n, gives:

$$\begin{aligned} E_n &= TE\left[X\right] - k - cT \\ &\times \left[E_{d+1}\left[\Psi\right]\left(g + \frac{hd}{d+1} + p\right) - \frac{ap}{n}\right. \\ &+ \left. (b+p)\int_0^{\frac{a}{n}} \left(\frac{a}{n} - \psi\right) P_{d+1}\left(\psi\right) d\psi\right]. \end{aligned} \tag{4.3}$$

Evidently, the residual energy depends on the coupling point, $\frac{a}{n}$, as well as on the PDF of the data transmission rate per sensor node, $P_{d+1}\left(\psi\right)$. In the remainder of this section, different cases for $P_{d+1}\left(\psi\right)$ are considered to derive the residual energy under different statistical characterisations for the data transmission rate of each node and examine the conditions under which $E_n = 0$ (i.e., energy neutrality), is achieved.

4.3.1 Illustrative Case: Uniform Distribution

When no knowledge of the underlying statistics of the data generation process exists, one can assume that $P_{d+1}\left(\psi\right)$ is uniform over the interval $[0, 2(d+1)r]$:

$$P_{d+1,\mathrm{U}}\left(\psi\right) = \begin{cases} \frac{1}{2(d+1)r}, & 0 \leq \psi \leq 2(d+1)r \\ 0, & \text{otherwise} \end{cases}. \tag{4.4}$$

The expected value of Ψ is $E_{d+1,\mathrm{U}}\left[\Psi\right] = (d+1)r$ bps. If $\frac{a}{n} > 2(d+1)r$, then the sink coupling point is always overprovisioned; thus, each node will remain in idle state consuming energy for beaconing and radio on, which cannot lead to optimal energy efficiency. This case is not

detailed here. For $\frac{a}{n} \leq 2(d+1)r$, by using (4.4) in (4.3), the following is obtained:

$$\begin{aligned} E_{n,\mathrm{U}} &= TE[X] - k - cT \\ &\times \left[(d+1)\, r \left(g + \frac{hd}{d+1} + p\right)\right. \\ &- \left.\frac{ap}{n} + \frac{a^2\,(b+p)}{4(d+1)rn^2}\right] \end{aligned} \tag{4.5}$$

If $p = 0$ then $E_{n,\mathrm{U}}$ is monotonically increasing with n as there is no energy penalty for buffering data and the optimal number of nodes is (trivially) infinity. Moreover, if $b = p = 0$, then (4.3) is independent of n as this assumes no energy penalties. Given that these cases lead to trivial solutions, they are not investigated further. For $b, p \neq 0$, the first derivative of $E_{n,\mathrm{U}}$ to n is

$$\frac{dE_{n,\mathrm{U}}}{dn} = cT\left[-\frac{ap}{n^2} + \frac{a^2\,(b+p)}{2(d+1)rn^3}\right]. \tag{4.6}$$

Under a given two-tier cluster-tree topology for $n \in (0, \infty)$, the number of nodes for which $\frac{dE_{n,\mathrm{U}}}{dn} = 0$ is[2]

$$n_{0,\mathrm{U}} = \frac{a\,(b+p)}{2p(d+1)r} \tag{4.7}$$

As (4.7) is the only admissible solution of $\frac{dE_{n,\mathrm{U}}}{dn} = 0$ for $n \in (0, \infty)$, $n_{0,\mathrm{U}}$ is the global extremum or inflection point of$E_{n,\mathrm{U}}$. The second derivative of $E_{n,\mathrm{U}}$ is

$$\frac{d^2E_{n,\mathrm{U}}}{dn^2} = cT\left[\frac{2ap}{n^3} - \frac{3a^2\,(b+p)}{2(d+1)rn^4}\right]. \tag{4.8}$$

Evaluating $\frac{d^2E_{n,\mathrm{U}}}{dn^2}$ for $n_{0,\mathrm{U}}$ nodes, gives

$$\frac{d^2E_{n,\mathrm{U}}}{dn^2}\left(n_{0,\mathrm{U}}\right) = -\frac{8cT\,(d+1)^3\,p^4r^3}{a^2\,(b+p)^3}, \tag{4.9}$$

which is negative (since all the variables are positive). Thus, the maximum possible residual energy for $n \in (0, \infty)$ is achieved under $n = n_{0,\mathrm{U}}$, and it is:

$$\begin{aligned} \max\{E_{n,\mathrm{U}}\} &= TE[X] - k - cT\,(d+1)\,r \\ &\times \left[g + \frac{hd}{d+1} + \frac{pb}{b+p}\right]. \end{aligned} \tag{4.10}$$

The last equation demonstrates that the maximum residual energy obtained is zero–balanced

[2]When used in a practical setting, the optimal value for the number of nodes must be rounded to the nearest integer. However, for exposition simplicity, rounding is not explicitly indicated in this notation.

consumption and production can be achieved over the harvesting interval, under energy harvesting with rate given by:

$$\begin{aligned} \min\left\{E\left[X\right]\right\}_{\mathrm{U}} &= \frac{k}{T} + c\left(d+1\right)r \\ &\times \left(g + \frac{hd}{d+1} + \frac{pb}{b+p}\right). \end{aligned} \tag{4.11}$$

Hence, if the energy harvester of the node achieves at least $\min\left\{E\left[X\right]\right\}_{\mathrm{U}}$ W (averaged over the interval of T seconds), this suffices for perpetual (energy neutral) operation of a WSN comprising $n_{0,\mathrm{U}}$ nodes at the same tier of the cluster-tree topology, with each node transmitting data with uniform rate between $[0, 2(d+1)r]$ bps. The minimum power shown in (4.11) is obtained under the operational parameters: c, T, d, k, g, h, b, p (see Table 4.1), $n_{0,\mathrm{U}}$ nodes and $E\left[\Psi\right] = r$. These parameters can be found for the specific technology used and the application requirements, as shall be shown in Sections 4.4 and 4.5.

The value derived for $n_{0,\mathrm{U}}$ by (4.7) is a real number. Within a practical setting, one should select $\lfloor n_{0,\mathrm{U}} \rfloor$ (if greater than zero) or $\lceil n_{0,\mathrm{U}} \rceil$, depending on which one derives the highest residual energy value in (4.5). Since the minimum harvested power required for energy neutral operation and the number of nodes achieving it have a critical dependence on the data transmission rate and its characteristics, the next subsection derives results under different characterisations for Ψ that are encountered often in practical data gathering applications based on WSNs and IoT applications. Similarly as for this subsection, once the result for the continuous case is derived, one can immediately derive the discrete case equivalent by converting the optimal value of n to the nearest integer that provides for the highest residual energy.

4.3.2 Examples of Analytic Derivation of Minimum Harvesting Power to Sustain Energy Neutrality

The previous calculation can now be generalised to other distributions expressing commonly observed data transmission rates in practical applications. Three additional PDFs for Ψ that have been used to model the marginal statistics of many real-world data transmission applications are considered, with the obtained analytic results in this subsection. Since the proofs follow the same process as for the uniform distribution, they are given in Appendix 7.1 in summary form. For each distribution, the parameters are coupled to the average transmission rate of the uniform distribution, $(d+1)\,r$, such that it is possible to achieve the same average data transmission rate over any uniformly-formed WSN cluster-tree topology where each node relays data from d additional nodes. This facilitates comparisons of the minimum harvesting

capability required under different characterisations for data rate.

4.3.2.1 Pareto distribution and fixed data rate

This distribution has been used, amongst others, to model the marginal data size distribution of TCP sessions that contain substantial number of small files and a few very large ones [137, 138]. Consider $P_{d+1,\mathrm{P}}(\psi)$ as the Pareto distribution with scale v and shape $\alpha \geq 2$ ($\alpha \in \mathbb{N}$),

$$P_{d+1,\mathrm{P}}(\psi) = \begin{cases} \alpha \frac{v^{\alpha}}{\psi^{\alpha+1}}, & \psi \geq v \\ 0, & \text{otherwise} \end{cases}. \tag{4.12}$$

The expected value of Ψ is $E_{d+1,\mathrm{P}}[\Psi] = \frac{\alpha v}{\alpha - 1}$ bps. Thus, setting

$$v = \frac{\alpha - 1}{\alpha}(d+1)r \tag{4.13}$$

gives $E_{d+1,\mathrm{P}}[\Psi] = (d+1)r$ bps, i.e., it matches the expected data transmission rate to that of the Uniform distribution. For the case of the Pareto distribution, if $\frac{a}{n} < v$, this corresponds to each node always attempting to transmit more data than what is allowed by the coupling point. This case will always incur energy penalty for buffering the residual bits beyond the coupling point and it is thus not investigated further as it will not lead to an optimal solution. For $\frac{a}{n} \geq v$, and (4.3) gives:

$$\begin{aligned} E_{n,\mathrm{P}} &= TE[X] - k - cT \left[\alpha v \frac{g + \frac{hd}{d+1} + p}{\alpha - 1} \right. \\ &+ \left. \frac{ab}{n} + (b+p)\left(\frac{v^{\alpha} n^{\alpha-1}}{a^{\alpha-1}(\alpha-1)} - \frac{\alpha v}{\alpha - 1} \right) \right]. \end{aligned} \tag{4.14}$$

Since $b + p \neq 0$, the number of nodes that derives the minimum power from the harvester to allow for energy neutrality under data transmission rate following the Pareto distribution of (4.12) is

$$n_{0,\mathrm{P}} = \frac{a}{v}\left(\frac{b}{b+p} \right)^{\frac{1}{\alpha}} \tag{4.15}$$

The minimum harvested power required under (4.15) is:

$$\begin{aligned} \min\{E[X]\}_{\mathrm{P}} &= \frac{k}{T} + c(d+1)r \left[g + \frac{hd}{d+1} \right. \\ &- \left. b + b^{\frac{\alpha-1}{\alpha}} (b+p)^{\frac{1}{\alpha}} \right] \end{aligned} \tag{4.16}$$

A special case for this distribution is when $\alpha = r$, which leads to $v = (d+1)(r-1)$ from (4.13). Then, the expected value of Ψ is $E_{d+1,\mathrm{F}}[\Psi] = (d+1)r$ bps and its standard deviation is $\sigma_{d+1,\mathrm{F}}[\psi] = (d+1)\sqrt{\frac{r}{r-2}}$. For $r > 150$ bps, the standard deviation is less than 0.7% of the mean value. Thus, in practice this case corresponds to transmission with fixed rate of $(d+1)r$ bps. This scenario occurs in WSNs capturing and transmitting data with fixed rate during their active time, e.g., in periodic temperature or humidity measurements gathered by a mote reading from a sensor [133, 132]. For this case, the number of nodes leading to the minimum harvested power is:

$$n_{0,\mathrm{F}} = \frac{a}{(d+1)(r-1)}\left(\frac{b}{b+p}\right)^{\frac{1}{r}} \tag{4.17}$$

For the vast majority of values for a, d and r used in practical WSN applications, $n_{0,\mathrm{F}}$ is equal to $\left\lfloor \frac{a}{(d+1)r} \right\rfloor$ (if greater than zero) or $\left\lceil \frac{a}{(d+1)r} \right\rceil$ when converted into an integer. This agrees with the intuitive answer for balancing fixed-rate transmission with $(d+1)r$ bps to consumption rate of a bps. The minimum harvested power required under (4.17) is:

$$\begin{aligned} \min\{E[X]\}_{\mathrm{F}} &= \frac{k}{T} + c(d+1)r\left[g + \frac{hd}{d+1}\right. \\ &- \left. b + b^{\frac{r-1}{r}}(b+p)^{\frac{1}{r}}\right] \end{aligned} \tag{4.18}$$

4.3.2.2 Exponential distribution

The marginal statistics of MPEG video traffic have often been modelled as exponentially decaying [139]. Consider $P_{d+1,\mathrm{E}}(\psi)$ as the Exponential distribution with rate parameter $\frac{1}{(d+1)r}$

$$P_{d+1,\mathrm{E}}(\psi) = \frac{1}{(d+1)r}\exp\left(-\frac{1}{(d+1)r}\psi\right) \tag{4.19}$$

for $\psi \geq 0$. In this case, the expected value of Ψ is $E_{d+1,\mathrm{E}}[\Psi] = (d+1)r$ bps. Via (4.3), one obtains

$$\begin{aligned} E_{n,\mathrm{E}} &= TE[X] - k - cT\left[(d+1)r\left(g + \frac{hd}{d+1} + p\right)\right. \\ &+ \frac{ab}{n} + (d+1)r(b+p) \\ &\times \left.\left[\exp\left(-\frac{a}{n(d+1)r}\right) - 1\right]\right]. \end{aligned} \tag{4.20}$$

Assuming $b \neq 0$, the value of

$$n_{0,\mathrm{E}} = \frac{a}{(d+1)r\ln\left(\frac{b+p}{b}\right)} \tag{4.21}$$

is the number of nodes that requires the minimum power from the harvester to allow for the system to maintain energy neutrality under data transmission following the exponential distribution of (4.19). The minimum harvested power required under this number of nodes is:

$$\begin{aligned} \min\left\{E\left[X\right]\right\}_{\mathrm{E}} &= \frac{k}{T} + c\left(d+1\right)r \\ &\times \left[b\ln\left(\frac{b+p}{b}\right) + g + \frac{hd}{d+1}\right]. \end{aligned} \tag{4.22}$$

4.3.2.3 Half-Gaussian distribution

This subsection is concluded by considering $P_{d+1,\mathrm{H}}\left(\psi\right)$ as the Half-Gaussian distribution with mean $E_{d+1,\mathrm{H}}\left[\Psi\right] = \left(d+1\right)r$

$$P_{d+1,\mathrm{H}}\left(\psi\right) = \begin{cases} 0, & \psi < 0 \\ \frac{2}{\pi(d+1)r}\exp\left(-\frac{\psi^2}{\pi(d+1)^2r^2}\right), & \psi \geq 0 \end{cases} \tag{4.23}$$

This distribution has been used widely in data gathering problems in science and engineering when the modelled data has non-negativity constraints. Some recent examples include the statistical characterisation of motion vector data rates in Wyner-Ziv video coding algorithms suitable for WSNs [130], or the statistical characterisation of sample amplitudes captured by an image sensor [135, 140]. Via (4.3), one obtains

$$\begin{aligned} E_{n,\mathrm{H}} &= TE\left[X\right] - k - cT\left[(d+1)r\left(g + \frac{hd}{d+1} + p\right)\right. \\ &- \frac{ap}{n} + (b+p)\left[(d+1)r\left[\exp\left(-\frac{a^2}{\pi(d+1)^2r^2n^2}\right)\right.\right. \\ &- \left.\left.1\right] + \frac{a}{n}\mathrm{erf}\left(\frac{a}{\sqrt{\pi}(d+1)rn}\right)\right]\Bigg], \end{aligned} \tag{4.24}$$

with $\mathrm{erf}\left(\cdot\right)$ the error function that can be approximated by its Taylor series expansion. Under $b \neq 0$ and $p \neq 0$, the number of nodes that leads to the minimum power required from the harvester in order for the system to maintain energy neutrality under data transmission rate (per node) characterised by $P_{d+1,\mathrm{H}}\left(\psi\right)$ is

$$n_{0,\mathrm{H}} = \frac{a}{\sqrt{\pi}(d+1)r\mathrm{erf}^{-1}\left(\frac{p}{b+p}\right)}, \tag{4.25}$$

with $\text{erf}^{-1}(\cdot)$ the inverse error function, which can be approximated by its series expansion. The minimum harvested power required under (4.25) is:

$$\begin{aligned} \min\{E[X]\}_{\text{H}} &= \frac{k}{T} + c(d+1)r\left[g + \frac{hd}{d+1} - b\right. \\ &+ \left.(b+p)\exp\left(-\left[\text{erf}^{-1}\left(\frac{p}{b+p}\right)\right]^2\right)\right]. \end{aligned} \tag{4.26}$$

4.3.3 Considering the Relay Case under a Multi-hop Topology

When expanding this analysis to multi-layer topologies, one can consider a variety of settings as illustrated in Figure 4.1. Here there are three distinguishable cases:

Firstly, when each node shapes its overall data transmission rate (which includes their own data and the data received from other nodes) according to one of the distributions considered in the previous subsection, the results will follow what was discussed before.

Secondly, when each node simply aggregates the received data with its own data within each transmission opportunity, thereby leading to a new data production rate PDF, one must consider this new distribution in the proposed analytic framework. Such distributions will be the convolutions of identical Uniform, Pareto, Exponential and Half-Gaussian distributions. For small values of d, e.g., $1 \leq d \leq 3$, the results can be derived following the steps given in Subsection 4.3.1 and 4.3.2 if functions

$$P_{d+1,\text{Z}}(\psi) = \underbrace{P_{\text{Z}}(\psi) \star \ldots \star P_{\text{Z}}(\psi)}_{d \text{ times}}, \ \text{Z} \in \{\text{U}, \text{P}, \text{E}, \text{H}\}$$

are derived. Given that $P_{d+1,\text{Z}}(\psi)$ and the $\int_0^{\frac{a}{n}} \left(\frac{a}{n} - \psi\right) P_{d+1,\text{Z}}(\psi)\, d\psi$ term of (4.3) can be computed with the help of a numerical package (e.g., Mathematica or MATLAB Symbolic) and that these will vary for each value of d, these cases are not expanded on further.

Finally, when $d \geq 4$, according to the central limit theorem [141], all data rate PDFs will begin to converge to a Gaussian distribution. By considering $P_{d+1,\text{N}}(\psi)$ as the Gaussian distribution with mean $E_{d+1,\text{N}}[\Psi] = (d+1)r$ and standard deviation σ

$$P_{d+1,\text{N}}(\psi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\psi - (d+1)r)^2}{2\sigma^2}\right), \tag{4.27}$$

via (4.3), the following is obtained (see Appendix 7.1.4 for details on this derivation):

$$
\begin{aligned}
E_{n,\mathrm{N}} &= TE\left[X\right] - k - cT\Bigg[\left(d+1\right)r\left(g+\frac{hd}{d+1}+p\right) \\
&- \frac{ap}{n} + \left(b+p\right)\Bigg[\frac{n\left(d+1\right)r-a}{2n} \\
&\times \left[\mathrm{erf}\left(\frac{\left(d+1\right)r-\frac{a}{n}}{\sqrt{2}\sigma}\right) - \mathrm{erf}\left(\frac{\left(d+1\right)r}{\sqrt{2}\sigma}\right)\right] \\
&+ \frac{\sigma}{\sqrt{2\pi}}\Bigg[\exp\left(\frac{-\left(\left(d+1\right)r-\frac{a}{n}\right)^2}{2\sigma^2}\right) \\
&- \exp\left(\frac{-\left(\left(d+1\right)r\right)^2}{2\sigma^2}\right)\Bigg]\Bigg]\Bigg].
\end{aligned}
\tag{4.28}
$$

The residual energy of (4.28) has a global maximum for $n \in (0,\infty)$, i.e., a global minimum in the required harvesting power $E\left[\chi\right]$, if: *(i)* $b \neq 0$ or $p \neq 0$ and *(ii)* the following condition is satisfied:

$$
\left|\mathrm{erf}\left(\frac{\left(d+1\right)r}{\sqrt{2}\sigma}\right) - \frac{2p}{b+p}\right| < 1. \tag{4.29}
$$

Then, the number of nodes that leads to the minimum power required in order for the system to maintain energy neutrality under data transmission (per node) following $P_{d+1,\mathrm{N}}\left(\psi\right)$ is

$$
n_{0,\mathrm{N}} = \frac{a}{\left(d+1\right)r - \sqrt{2}\sigma c_{\mathrm{N}}}, \tag{4.30}
$$

with $\left(d+1\right)r \neq \sqrt{2}\sigma c_{\mathrm{N}}$,

$$
c_{\mathrm{N}} = \mathrm{erf}^{-1}\left(\mathrm{erf}\left(\frac{\left(d+1\right)r}{\sqrt{2}\sigma}\right) - \frac{2p}{b+p}\right). \tag{4.31}
$$

The minimum harvested power required under (4.30) is:

$$
\begin{aligned}
\min\left\{E\left[X\right]\right\}_{\mathrm{N}} &= \frac{k}{T} + c\left(d+1\right)r\Bigg[g+\frac{hd}{d+1} \\
&+ \frac{\sigma\left(b+p\right)}{\sqrt{2\pi}\left(d+1\right)r}\left[\exp\left(-c_{\mathrm{N}}^2\right)\right. \\
&- \left.\exp\left(-\frac{\left(\left(d+1\right)r\right)^2}{2\sigma^2}\right)\right]\Bigg].
\end{aligned}
\tag{4.32}
$$

4.3.4 Discussion

The results of this section can be used in practical applications to assess the impact in the required harvesting power and time when the statistics of the transmission data rate follow a

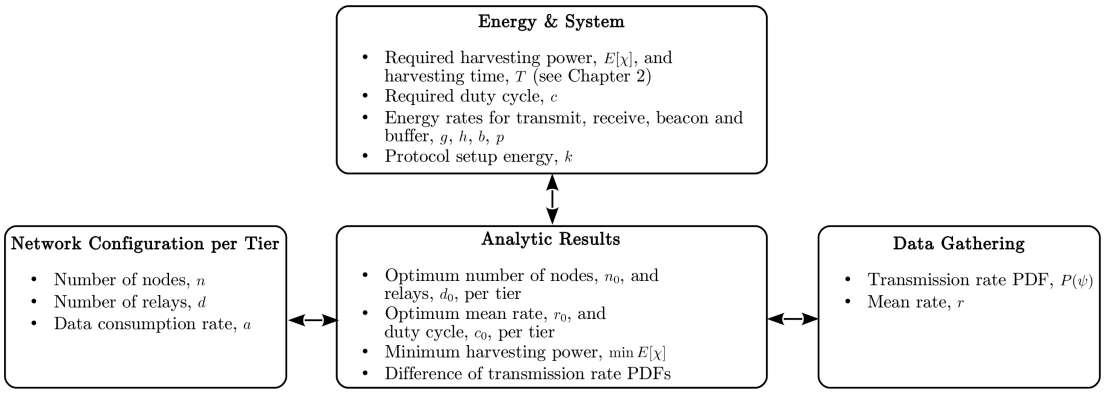


Figure 4.3: Conceptual illustration of the links between: network, system and data gathering via the proposed analysis. When parameters from two out of three domains are provided, the analytic framework can tune the parameters of the third. The symbol definitions are provided in Table 4.1.

certain PDF and the network parameters are fixed. Conversely, if a particular technology, such as an array of photovoltaic cells or a piezoelectric transducer, has been shown to provide for certain power generation capability per sensor (such as in Chapter 3), under the knowledge of the system and data gathering parameters and the duty cycle of the network, one can establish the appropriate network parameters per tier. Finally, for given network and system parameters, one can assess the achievable data transmission rates such that the network infrastructure remains energy neutral.

Thus, as shown in Figure 4.3, the analytic results allow for the linking of network, data gathering, and energy and system parameters within uniformly-formed cluster-tree node topologies. Hence, this analysis can be used for early stage exploration of the capabilities of a particular WSN or IoT infrastructure in conjunction with the data gathering requirements of a particular application, prior to embarking in cumbersome development and testing in the field.

4.4 Evaluation of the Analytic Results

4.4.1 WSN and System Settings

A typical WSN setup comprising of several TelosB nodes (using the IEEE 802.15.4 standard [24] with the CC2420 transceiver [29]) are used to make a testbed, with nodes running the low-power Contiki-OS 2.6 IoT operating system. All nodes use the TFDMA protocol [31] to communicate with the base station existing on the same channel, following topologies such as the ones shown in Figure 4.1. The TFDMA protocol uses biology-inspired self-maintaining algorithms in wireless sensor nodes and achieves near optimum TDMA characteristics in a decentralised manner and over multiple channels (frequencies). This is achieved by extending the concept of collaborative reactive listening in order to balance the number of nodes in all 16 available

channels of IEEE 802.15.4 2.4 GHz PHY [31]. Consequently, TFDMA can be deployed at the application layer with very low complexity and provides for balanced multichannel coordination of multiple nodes. Its use allows for quick convergence to the steady-state and permits collision-free communications once steady-state has been established. It also provides for comparable or superior bandwidth utilisation to channel-hopping approaches like TSMP [92], TSCH [1] and EM-MAC [3]. However, similar results can be obtained with any other protocol offering collision-free communications under a single- or multiple-channel cluster-tree topology, such as TSMP, IEEE 802.15.4 GTS [118, 63], and so on.

Under TFDMA, with an active time $T_{\text{act}} = 400$ s, convergence has been shown to occur in less than 1.3% of T_{act} (3–5s) and, on average, the energy dissipation for convergence has been found to be $k = 165.6$ mJ for the TelosB nodes used in the testbed. Concerning the communications side, following the default TFDMA setup, the packet size was set to 114 bytes, the DESYNC interval to 1 s and the DESYNC coupling constant to 0.95 [32] for all measurements. Each node transmits 1-byte beacon packets every 8 ms when not transmitting data packets during its transmission slot, maintaining connectivity and synchronisation. Finally, since the TFDMA protocol ensures no collisions occur during the steady-state active mode, the very-low complexity NullMAC and NullRDC options of Contiki-OS are used, which lead to maximum data consumption rate at the application layer of $a = 144$ kbps.

Concerning the data gathering itself, artificial data is created via a custom MATLAB function that, starting from the `rand()` function, generates data with Uniform, Pareto, Exponential and Half-Gaussian distributions (considered in Section 4.3) via rejection sampling [142], with mean transmission rate equal to $r = 24$ kbps. The data is copied onto each node and it is read from its external flash memory during the steady-state active mode. This ensures that: *(i)* the different PDFs under consideration are matched and *(ii)* the energy to retrieve this data from the flash memory replaces the sensing and processing energy that would have been dissipated if the data had come from an actual sensing process.

Under these operational settings, the energy measurement setup comprises a high-tolerance 1 Ohm resistor placed in series with each TelosB node. Knowing that each node operates at 3 Volt and measuring the potential difference across the resistor (and therefore current flow through the resistor), the real-time energy consumption (see Figure 4.2 for examples) can be derived. The time resolution for the power measurements was 12.5 kHz using a Tektronix MDO4104-6 oscilloscope. Under this setup, it was also possible to measure the different energy rates of Table 4.1 by enabling transmission, listening, writing to flash memory and beaconing during the idle state to maintain synchronisation. They were found to be: $g = 2.29262 \times 10^{-7}$ J/b, $h = 2.92309 \times 10^{-6}$ J/b, $p = 3.89392 \times 10^{-7}$ J/b and $b = 2.17324 \times 10^{-7}$ J/b.

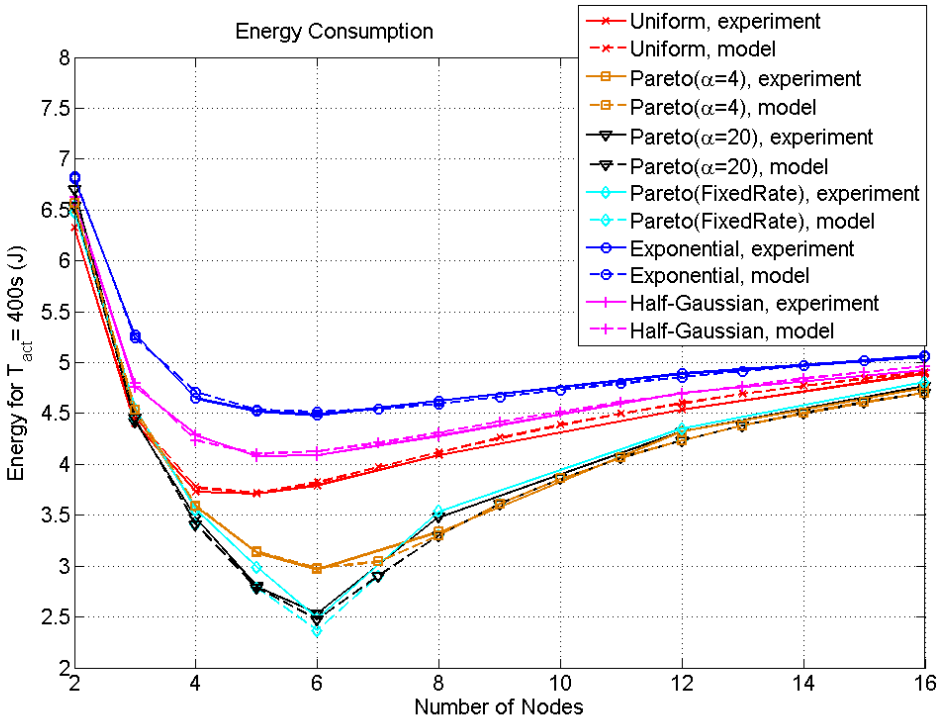


Figure 4.4: Energy consumption per node under different data transmission PDFs. The experiments correspond to $T_{\text{act}} = 400$ s, $k = 165.6$ mJ and $d = 0$.

4.4.2 Model Validation

Consider a fully connected (single hop) topology, as shown in Figure 4.1(left). Each node sends only its own data, which corresponds to $d = 0$ (no relay) and repeated tests with various values for n (total number of nodes within each channel) are undertaken. Given that TFDMA leads to balanced topologies within each channel, Figure 4.4 presents the results obtained by each node of one channel as all the remaining channels produce identical performance. For each data production PDF, the "theoretical" results have been produced via (4.5), (4.14), (4.20), (4.24) by considering only the energy dissipation part; the energy harvesting part, $TE[X]$, is discussed separately in Section 4.5.1. Evidently, for the vast majority of cases, the theoretical and experimental results are in agreement, with the maximum difference between them limited to within 0.237 J, i.e., a maximum error of 7.2% – the same level of accuracy was observed under a variety of tested rates, r, and TFDMA settings.

The results of Figure 4.4 demonstrate that each transmission rate distribution incurs different energy consumption. Thus, the manner the data traffic is shaped in a WSN plays an important role in the system's requirements for energy neutrality. Moreover, the results show that, depending on the data transmission rate PDF, the number of nodes where the minimum energy consumption occurs, i.e., $n_{0,\text{U}}$, $n_{0,\text{P}}$, $n_{0,\text{F}}$, $n_{0,\text{E}}$ and $n_{0,\text{H}}$, may differ. The accuracy of the analytic estimations is quantified in Table 4.2 in comparison to the experimentally obtained

Table 4.2: Differences in the minimum harvested energy required amongst the considered PDFs under the settings of Figure 4.4.

Considered PDFs	Theoretical difference (J)	Experimental difference (J)	Percentile error (%)
Pareto $\alpha = 4$ vs. Uniform	-0.729	-0.742	1.81
Pareto $\alpha = 20$ vs. Uniform	-1.229	-1.176	4.32
Fixed-Rate vs. Uniform	-1.339	-1.223	8.63
Exponential vs. Uniform	+0.803	+0.775	3.45
Half-Gaussian vs. Uniform	+0.394	+0.372	5.54
Half-Gaussian vs. Exponential	-0.409	-0.403	1.44

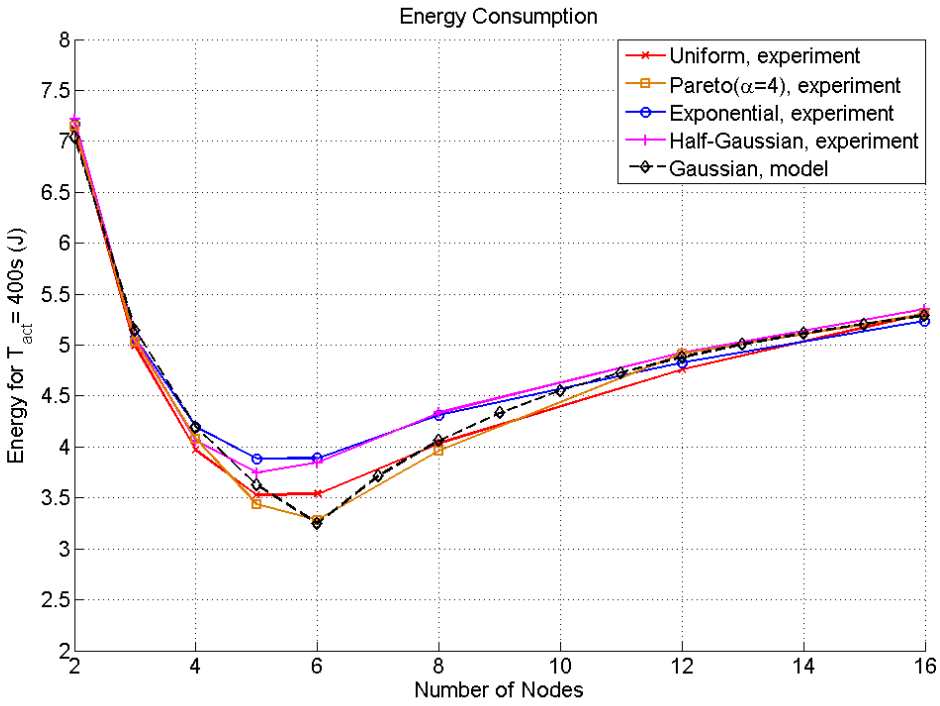


Figure 4.5: Energy consumption per node with different data production PDFs, $d = 4$ and $r = 4.8$ kbps; each node aggregates the received data with its own data within each transmission opportunity.

values for the difference in the minimum energy consumption. The table demonstrates that the theoretically calculated differences are very close to the experimentally obtained values, as the average percentile error is only 4.20%.

Finally, with respect to a multi-hop scenario, Figure 4.5 presents the results under $d = 4$ and $r = 4.8$ kbps (with all other settings being the same). As expected, all experimental curves converge towards the results of the Gaussian distribution. This convergence improves further when higher values of d are considered.

4.5 Applications

4.5.1 Maximising Active Time under Given Energy Harvesting Capability

The first application under consideration concerns networks where every sensor is equipped with certain energy harvesting technology, e.g., a piezoelectric unit or photovoltaic cells. Under a given data transmission rate PDF with mean r (fitted to the experimentally observed data rate histogram), the aim is to derive the optimal number of sensors (n_0) and the maximum duty cycle (c_0) so that the network performs data gathering and transmission for the maximum amount of time under energy neutrality. Such a scenario occurs in energy management systems for WSNs, indoor or outdoor monitoring systems that are expected to be active for the maximum amount of time possible [65, 22, 133] (see Section 2.2).

The expected power produced by harvesting, $E[X]$, for photovoltaic and piezoelectric technologies under stable indoor conditions (as reported in the relevant literature, see Chapter 3) and consider a single-tier network topology [Figure 4.1(a)]. The goal is to match the expected energy harvested within T s with the expected energy consumption within T_{act} s and report the highest possible values for the duty cycle, c, and active time, T_{act}.

For the data rate PDFs considered in this chapter and the system settings of 4.4, the obtained results in Table 4.3 under harvesting time $T = 21600$ s (6 hr) and $r = 3000$ bps. Under the given value for $TE[X]$, the values for n_0 were derived using: (4.7), (4.15), (4.17), (4.21) and (4.25); c_0 (and T_{act}) were derived solving: (4.11), (4.16), (4.18), (4.22) and (4.26) for c. Evidently, depending on the technology used and the chosen data transmission rate PDF, the results can vary, i.e., from energy neutrality achieved with $c_0 = 0.112$ and $T_{\text{act}} = 2424$ s for Exponentially distributed data gathering and transmission rate, to $c_0 = 0.279$ and $T_{\text{act}} = 6038$ s for Pareto distributed (or fixed) data gathering and transmission. For applications that require continuous monitoring ($c_0 = 1$), based on the results from Table 4.3 it is possible to calculate how many independent sets of n_0 nodes should be installed so that continuous monitoring is achieved under energy neutrality. For example, since $c_0 > 0.25$ under the Pareto PDF (with $\alpha \geq 20$) and piezoelectric harvesting, one can predicted that, by installing four independently operating sets of 48 nodes and imposing that only one set is active at any given time, constant monitoring & transmitting and energy neutrality is ensured under energy neutrality.

This framework allows for such studies to be done at early design stages and can incorporate all the relevant parameters of the WSN (protocol-related parameters, system settings, active time, and so on) in order to meet the requirements imposed by a given application.

Table 4.3: Maximum active time T_{act} and duty cycle c (in parentheses) required by two different harvesting technologies under harvesting time $T = 21600$ s (6 hr) and with mean data rate $r = 3000$ bps.

Production Rate PDF	16 cm^2 Solar Panel $E[X] = 160$ μW [37]	Piezoelectric Unit $E[X] = 200$ μW [39]
Uniform, $n_0 = 37$	2975 s (0.138)	3755 s (0.174)
Pareto ($\alpha = 4$), $n_0 = 50$	3754 s (0.173)	4729 s (0.219)
Pareto ($\alpha = 20$), $n_0 = 48$	4556 s (0.211)	5753 s (0.266)
Fixed rate, $n_0 = 48$	4782 s (0.221)	6038 s (0.279)
Exponential, $n_0 = 47$	2424 s (0.112)	3061 s (0.142)
Half-Gaussian, $n_0 = 42$	2677 s (0.124)	3380 s (0.156)

Table 4.4: Minimum harvesting requirement ($\min\{E[X]\}$) for ad hoc settings and optimised mean data rate and duty cycle adjustment with the proposed framework. The energy saving shows the percentile difference between the minimum harvesting requirement for the ad hoc and proposed cases. The network parameters of this example correspond to Figure 4.1(centre).

		Tier 1 $d = 0$, $n = 4$	Tier 2 $d = 2$, $n = 2$
Ad Hoc	Pareto ($\alpha = 4$)	$r_{\text{adhoc}} = 22000$ bps $c_{\text{adhoc}} = 0.0132$ $\min\{E[X]\} = 112$ μW	$r_{\text{adhoc}} = 12000$ bps $c_{\text{adhoc}} = 0.0241$ $\min\{E[X]\} = 735$ μW
	Fixed Rate	$r_{\text{adhoc}} = 22000$ bps $c_{\text{adhoc}} = 0.0132$ $\min\{E[X]\} = 108\mu$W	$r_{\text{adhoc}} = 12000$ bps $c_{\text{adhoc}} = 0.0241$ $\min\{E[X]\} = 728$ μW
Proposed	Pareto ($\alpha = 4$)	$r_0 = 37135$ bps $c_0 = 0.0078$ $\min\{E[X]\} = 87\mu$W	$r_0 = 24757$ bps $c_0 = 0.0117$ $\min\{E[X]\} = 594$ μW
	Fixed Rate	$r_0 = 36000$ bps $c_0 = 0.0080$ $\min\{E[X]\} = 68\mu$W	$r_0 = 24000$ bps $c_0 = 0.0121$ $\min\{E[X]\} = 539$ μW
Energy Saving (%)	Pareto ($\alpha = 4$)	22.32	19.18
	Fixed Rate	37.04	25.96

4.5.2 Minimising Power Harvesting Requirements under a Fixed Network Setup

In a second application example, a typical structural monitoring system is considered, such as the one proposed by Notay and Safdar [143]. In such systems, several sensors are embedded into a structure (e.g., sensors embedded within an aeroplane's wings or within the steel structure of a bridge) in order to gather and transmit measurements to collection points. The collection points relay measurements (along with their own) to Wi-Fi equipped access points [143] that have fixed power supply. The sensors harvest energy via the vibrations of the structure (e.g., aeroplane wing vibrations during flight) but energy neutrality must be ensured with the minimum possible power harvesting as the sensors are located in difficult to service areas and must be able to operate in perpetuity. In such applications there is no strict real-time constraint for the data collection, as a volume of V_{fixed} bytes of measurements is collected for batch offline analysis of

structural properties and reaction times are in the order of hours or even days. Finally, the harvesting time interval is imposed by the application context, e.g., the average duration of a flight or a bridge's structural vibrations occurring during peak usage hours each day. Thus, the mean data transmission rate can be adjusted to the setting minimising the required power harvesting under the pre-established network setup and harvesting time interval.

Under a given two-tier cluster-tree topology, such as shown in Figure 4.1(centre), with:

- fixed number of sensors and fixed relay configuration per tier ($n_0 \equiv n_{\text{fixed}}$,$d \equiv d_{\text{fixed}}$),
- fixed requirements for the harvesting time and the volume of data to be collected by each node, i.e.,

$$T \equiv T_{\text{fixed}} \text{ and } r \times c \times T_{\text{fixed}} \equiv V_{\text{fixed}}, \tag{4.33}$$

- the assumption of Pareto-distributed or fixed data transmission rate (i.e., $\alpha \equiv \alpha_{\text{fixed}}$),

the mean rate and duty cycle setting per tier that minimise the harvested power requirements are derived. This is achieved by: *(i)* deriving v_0 by solving (4.15) for v under $n_0 \equiv n_{\text{fixed}}$; *(ii)* deriving r_0 by solving (4.13) for r under $v \equiv v_0$ and $d \equiv d_{\text{fixed}}$; *(iii)* deriving $c_0 \equiv \frac{V_{\text{fixed}}}{r_0 T_{\text{fixed}}}$. This effectively "tunes" the duty cycle and the mean data rate so that the fixed network and data transmission settings listed above become optimal, i.e., they lead to the minimum power harvesting that ensures energy neutrality. Under the system settings of Section 4.4. and $V_{\text{fixed}} = 25$ Mbit and $T_{\text{fixed}} = 86400$ s, Table 4.4 shows the derived minimum power harvesting requirements in comparison to the results obtained under an ad hoc allocation of data rates and duty cycles per tier.

Both the ad hoc and the proposed settings satisfy the conditions imposed by (4.33) and lead to energy neutrality. However, deriving the mean rate and duty cycle per tier under the proposed framework meets these constraints with substantial savings in the required power harvesting, which were experimentally found to range between 19% to 37% in comparison to the ad hoc settings**.** Hence, under piezoelectric harvester producing $E[X] = 200\ \frac{\mu\text{W}}{\text{cm}^2}$ [39], the proposed approach requires an active harvesting area of 2.7–3.0 cm^2 per node for Tier 2 while the ad hoc approach requires 3.6–3.7 cm^2.

4.6 Conclusions

This chapter proposed an analytic framework for characterising practical energy neutrality in uniformly-formed WSNs. The framework recognises the importance of the application data transmission rate in the network's energy dissipation. Specifically, it provides for an analytic assessment of the expected energy dissipation in function of the system parameters, under a

variety of statistical characterisations for the data transmission rate of each sensor node. The experimental assessment on an energy measurement testbed of low-power TelosB nodes and recently-proposed, collision-free, communication protocol with rapid, low-energy, convergence validates the analytic framework matches experiments with accuracy that is within 7% of the measured energy consumption.

The framework presented in this chapter can be used in conjunction with particular harvesting technologies such as those presented in Chapter 3 to predict the smallest possible energy harvesting interval for an energy-neutral deployment before costly and cumbersome testing in the field. Finally, this analysis could be used in conjunction with future energy-harvesting IoT and WSN systems and technologies in order to predict the best possible data transmission rate that can be accommodated in function of the system's operational settings.

Thank you to Miss Hana Besbes for her contributions towards the mathematics in this chapter.

Chapter 5

Decentralised Time-Synchronised Channel Swapping for Ad Hoc Networks

5.1 Introduction

The recent thrust towards M2M communications [103, 104] and the integration of WSNs with the generic Internet infrastructure via 6LoWPAN support at the network layer [108, 105, 106] via newly envisaged IoT applications call for the development of ad hoc communication protocols at the MAC layer, i.e., protocols that do not depend on any preexisting infrastructure, such as fixed power access points, interference free control channels, and global time synchronisation between all nodes in the network. Such systems would rapidly become unreliable with large numbers of IoT devices interacting with the network, waking up for short periods to transmit data, and then returning to their sleep state.

The concept of channel hopping, introduced in Section 2.3.3, enables nodes to move between channels (frequencies) of the physical layer, and has gained acceptance as a good solution for wireless MAC layer coordination, since nodes are not constantly in a channel with excessive interference. TSCH [1] and multichannel DSRC [78] now comprise essential elements of the IEEE 802.15.4e-2012 [81, 82] and IEEE 802.11p [23] standards, respectively.

A study of related work on multichannel MAC protocols can be found in Section 2.3, specifically, Section 2.3.4, along with a working description of TSCH in Section 2.3.5.

Table 5.1 summarises the features of the most relevant multichannel MAC protocols in conjunction with the proposed DT-SCS protocol.

Table 5.1: Comparison of key features of existing MAC protocols vs. the proposed DT-SCS.

Protocol	IEEE 802.15.4e 2012 [81]	TSCH [1]	EM-MAC [3]	Proposed DT-SCS
Coordination	Centralised	Centralised	Distributed	Distributed
Multichannel Operation	Yes	Yes	Yes	Yes
Channel Hopping	No	Yes	Yes	Yes
Convergence Time (s)	~ 1	~ 14	~ 4	~ 1.5
Connectivity	Low	Medium	Low	High
Network Throughput (kbps)	$8-100$	$10-55$	$6-18$	$30-85$
Resilience to Interference	Low	High	High	High

Regarding the remainder of this chapter: Section 5.2 describes the proposed DT-SCS protocol. Section 5.3 analyses DT-SCS in terms of stability, connectivity, and convergence time. Section 5.4 presents simulation results, whereas Section 5.5 presents experiments with an ad hoc wireless network deployment. Finally, Section 5.6 concludes the chapter.

5.2 The Proposed DT-SCS Protocol

Section 5.2.1 presents an overview of the overall operation of the proposed DT-SCS protocol. The detailed operation of the SYNC and DESYNC mechanisms is given in the Section 5.2.2.

5.2.1 Introduction to the Basic Concept

Consider an ad hoc network comprising W wireless nodes randomly distributed in C channels [see the left part of Figure 5.1(a)], with each node broadcasting short beacon packets periodically every T seconds. Within each channel, nodes are assumed to be fully-connected (all nodes can hear all other nodes) or densely-connected (only a small subset of nodes cannot be reached by all nodes). The proposed DT-SCS balances the number of nodes per channel and adjusts the transmission time of each node's beacon packet to reach an evenly-distributed timeslot allocation within each channel [see the right part of Figure 5.1(a)]. Specifically, the nodes in each channel perform PCO-based desynchronisation (i.e., they are "DESYNC" nodes) and elect a single "SYNC" node to provide for cross-channel synchronisation. Within each period, the SYNC node of each channel listens for the SYNC beacon message in the next channel[1] and adjusts the transmission time of its own beacon packet in its own channel using PCO-based synchronisation

[1]Cyclic behaviour exists between the last and the first channel. For instance, in IEEE 802.15.4, the SYNC node of Channel 16 listens for the SYNC beacon message of Channel 1.

[33]. SYNC nodes will also move to the next channel if they detect that fewer nodes are present there. In this way, the network will converge to the steady state with $W_c = \frac{W}{C}$ nodes per channel[2]. The beacon packet transmission flow between DT-SCS nodes is illustrated in the right part of Figure 5.1(a).

Once the system reaches the steady state, SYNC or DESYNC nodes in adjacent channels can swap channels and timeslots in pairs using a simple RQ/ACK scheme. Figure 5.1(b) highlights the short interval between two consecutive beacon packet transmissions (stemming from two different nodes in a channel), during which RQ/ACK packet transmissions for channel swaps take place. If nodes join or leave the network, all remaining nodes adjust their beacon packet timings spontaneously, in order to converge to a new steady state. As shown in Figure 5.1(a), the key aspect of DT-SCS is the spontaneous convergence of the ad hoc wireless network from a random state to a multichannel time synchronised beaconing state, without the need for a coordinator node, a coordinating channel, or global time synchronisation.

Once convergence to steady-state is achieved, the only overhead in the proposed DT-SCS protocol stems from handling swap requests as well as beacon packet broadcasts. Both, however, are very short packets (less than ten bytes). Beyond this, the nodes can also be set to a "sparse listening" mode, as is detailed in the experimental section, Section 5.5. Therefore, the protocol overhead is minimal compared to data packet transmission and reception in data-intensive wireless networks.

Losses of beacon packets and timing errors due to interference cause node beacon times to waver, that is, nodes send beacon messages at incorrect times. As such, all nodes receiving these messages are similarly affected. If left untreated, this wavering could propagate through the network until all nodes are affected and the network is no longer considered to be converged. To combat this, the notion of coupling between nodes is introduced by PCOs [33, 119]: instead of a DESYNC node jumping directly to the midpoint of its beacon neighbours, the node gradually slides towards the mid point with coupling factor α ($0 < \alpha < 1$); this is also known as negative coupling in the PCO literature [33, 119, 144, 145]. Similarly, a SYNC node gradually adjusts its beaconing time by coupling factor β ($0 < \beta < 1$) to align with the beacon of the SYNC node in the next channel; this beacon alignment is also known as positive coupling [33, 146]. This work is the first to propose the usage of positive coupling for interchannel synchronisation in conjunction with concurrent intrachannel coordination achieved via negative coupling. As verified via simulations (Section 5.4) and experiments (Section 5.5), appropriate selection of coupling factors ensures that any noise and instability in beacon timings is attenuated and does not propagate uncontrollably throughout all nodes and channels of DT-SCS.

[2]For simplicity, it is assumed that W is divisible by C. However, when this is not the case the scheme balances the number of nodes to $W_c \in \left\{ \left\lfloor \frac{W}{C} \right\rfloor, \left\lceil \frac{W}{C} \right\rceil \right\}$ nodes per channel.

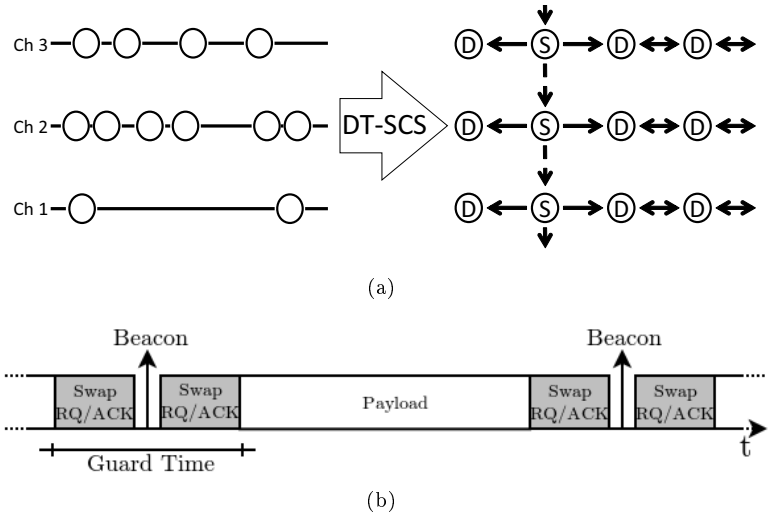


Figure 5.1: (a, left) Initial random state of $W = 12$ node in $C = 3$ channels; (a, right) DT-SCS converged state with $W_c = 4$ nodes per channel, showing the intrachannel desynchronisation (solid horizontal lines) and interchannel synchronisation (dashed vertical lines) between DESYNC (D) and SYNC (S) nodes, respectively. Arrows indicate the intended recipient of each beacon packet transmission. (b) The grey slots indicate the short transmitting/listening intervals where nodes can request and acknowledge swaps.

5.2.2 New Multichannel Coupling via Joint SYNC-DESYNC

Synchronisation and desynchronisation primitives are algorithms for revising the beacon packet broadcast time of a node in a wireless network based on the broadcast times of beacon packets from other nodes within a certain time interval. The proposed DT-SCS protocol is the first to combine SYNC and DESYNC algorithms in a joint framework for a decentralised collision-free multichannel MAC. Consider W_c nodes being present in channel c, with $c \in \{1, ..., C\}$, and the total nodes given by $W = \sum_{c=1}^{C} W_c$. Nodes join the network by broadcasting an initial beacon randomly in channel c at a time between $[0, T)$ seconds. Nodes repeat the transmission of their beacon upon the completion of their cycle, every T seconds. The fraction of the way through a cycle at a given time $t \in [0, T)$ is denoted as the node's phase [33, 119], $\varphi \in [0, 1)$.

As shown in Figure 5.2, the beacon packet transmission times can be seen as beads moving clockwise on a ring with period $T = 1$ s [32]. When the phase of a node becomes one (i.e., the bead reaches the top of the ring in Figure 5.2), a beacon packet is broadcast, and the node's phase is reset to zero. Each node keeps the phase of received beacon packets and updates its own beacon phase $\varphi_{\text{curr}}^{(k-1)}$ to $\varphi_{\text{curr}}^{(k)}$ based on the reactive listening primitive. Thus, superscript (k) indicates the kth phase-update iteration.

For the SYNC and DESYNC algorithms, it is immaterial which physical sensor node is linked to which beacon broadcast, as the phase update process is solely dependent on the received beacon packet times [32, 33, 119, 144, 145]. For this reason, this text explicitly refers to beacon packet transmission events and not the physical nodes that transmit them.

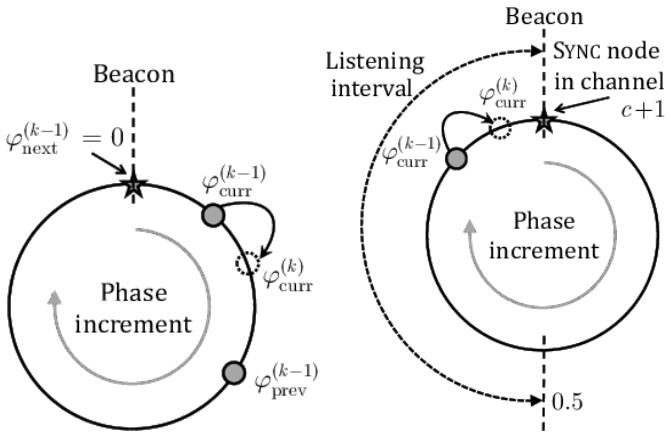


Figure 5.2: (a) A DESYNC node performs its kth phase update when the next DESYNC or SYNC beacon packet is received in channel c. (b) A SYNC node performs its kth phase update when a SYNC beacon packet is received in channel $c+1$ while the phase of the current beacon broadcast is within its listening interval.

5.2.2.1 DESYNC Phase Update via Negative Coupling

During desynchronisation in channel c, each node's beacon phase is updated once within each period T. As shown in Figure 5.2(a), the phase of node "curr" (current) is updated based on the phases of received "prev" (previous) and "next" beacon messages, originating from nodes that transmitted their beacon before and after currently considered ("curr") node, respectively. Specifically, upon receiving the next beacon packet, the phase of node "curr" moves towards the middle of the interval between the phases of "prev" and "next" beacon messages, i.e., the phase values of the nodes become decoupled. The kth phase update of DESYNC with such negative coupling is expressed by[3] [32, 144]

$$\varphi_{\text{curr}}^{(k)} = (1-\alpha)\,\varphi_{\text{curr}}^{(k-1)} + \frac{\alpha}{2}\left(\varphi_{\text{prev}}^{(k-1)} + \varphi_{\text{next}}^{(k-1)}\right) \mod 1, \tag{5.1}$$

with $\alpha \in (0,1)$ the DESYNC phase coupling constant controlling the speed of the phase adaptation and mod 1 denoting the modulo operation with respect to unity. Previous work [32, 144] showed that the reactive listening primitive of (5.1) disperses all beacon packet broadcasts in each channel $c \in \{1,...,C\}$ at intervals of $\frac{T}{W_c}$. This leads to fair TDMA scheduling in channel c in the steady state (SS).

After k_{ss} iterations of (5.1), all beacon packets in channel c are periodic and the phase

[3] Since (5.1) is applied when the next beacon packet is received, then $\varphi_{\text{next}}^{(k-1)} = 0$ [see Figure 5.2(a)]. However, the $\varphi_{\text{next}}^{(k-1)}$ term from (5.1) is included to clarify that the operation of DESYNC depends on both the previous and next beacon packet phase.

updates lead to convergence to SS, expressed by

$$\left|\varphi_{\text{curr}}^{(k_{\text{ss}})} - \varphi_{\text{curr}}^{(k_{\text{ss}}-1)}\right| \leq b_{\text{thres}}, \tag{5.2}$$

with b_{thres} the preset convergence threshold, typically $b_{\text{thres}} \in [0.001, 0.100]$. In the steady state of the DT-SCS protocol, each node in channel c transmits data packets for $T\left(\frac{1}{W_c} - b_{\text{thres}}\right) - t_{\text{swap}}$ seconds in the centre of its timeslot, where t_{swap} denotes the total duration of the guard time per node. Therefore, the maximum number of nodes supported under collision-free TDMA per channel c is less than $\left\lfloor \frac{1}{b_{\text{thres}}} \right\rfloor$.

5.2.2.2 Sync Phase Update via Positive Coupling

PCO-based synchronisation with positive coupling [33] updates each Sync node's beacon phase according to a received beacon packet (from another Sync node) that is within the listening interval $[\frac{T}{2}, T)$ (second half of the beaconing cycle) [see Figure 5.2(b)]. Under the proposed DT-SCS protocol, the phase of each Sync beacon in channel c changes after a Sync beacon packet is received in channel $c+1$ within the listening interval. Specifically, it moves closer to the phase of the node that sent the beacon packet in channel $c+1$. Hence, the kth phase update of PCO synchronisation [33] is performed at $\varphi_{\text{curr}}^{(k-1)}T$ s after the node's last beacon packet transmission, $0.5 < \varphi_{\text{curr}}^{(k-1)} < 1$, via the positive coupling:

$$\varphi_{\text{curr}}^{(k)} = (1+\beta)\,\varphi_{\text{curr}}^{(k-1)} \pmod 1, \tag{5.3}$$

with $\beta \in (0,1)$ the Sync phase coupling constant controlling the speed of the phase adaptation. Any beacon packets transmitted outside the listening interval $(0.5, 1)$ are ignored with respect to the Sync phase update. However, in the proposed DT-SCS, these packets are still processed to extract useful information, such as the total number of nodes in the current channel (see Section 5.2.3). After $\tilde{k}_{\text{ss}}$ phase updates, (5.3) converges to coordinated Sync beacon packet broadcasts at intervals of $(1 \pm \tilde{b}_{\text{thres}}) \times T$ seconds [33]. Similar to the Desync case, $\tilde{b}_{\text{thres}}$ is used[4] to detect convergence to the steady state under (5.2).

5.2.3 Proposed DT-SCS Protocol Description

In an ad hoc wireless network comprising W nodes that apply the Sync and Desync algorithms within C channels ($C > 1$), throughput is equally balanced across all nodes when the number of nodes is balanced across all channels, that is, when $W_c = \{\lfloor \frac{W}{C} \rfloor, \lceil \frac{W}{C} \rceil\}$ nodes are present

[4] The thresholds b_{thres} and $\tilde{b}_{\text{thres}}$ for the respective cases of Desync and Sync can have different values. For simplicity, in this implementation, b_{thres} and $\tilde{b}_{\text{thres}}$ are considered equal, $b_{\text{thres}} = \tilde{b}_{\text{thres}}$.

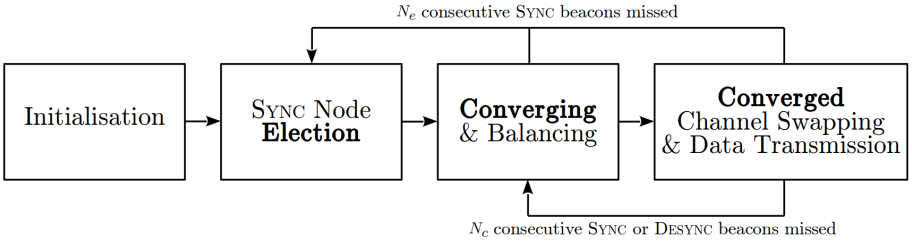


Figure 5.3: Block diagram of the operational modes of DT-SCS. The values of N_e and N_c are set via experimentation with varying packet loss. Data transmission and channel swapping takes place only during the `Converged` mode.

within each channel $c \in \{1, ..., C\}$. Figure 5.3 presents the basic stages of the proposed DT-SCS protocol, which are explained in the following subsections.

5.2.3.1 Node Initialisation and Beacon Packet Contents

When initialised, each node joins a channel $c \in \{1, ..., C\}$ randomly as a DESYNC node. Initially, nodes have their receivers constantly enabled and send their beacon messages according to the DESYNC rules.

Each beacon packet transmitted by each node in channel c contains:

1. the originating node type (SYNC or DESYNC);
2. the node unique identity number (node ID);
3. the node ID of the SYNC node in channel c (NULL if none);
4. the number of unique nodes heard in channel c, W_c;
5. the number of unique nodes heard (directly or indirectly) in channel $c+1$, W_{c+1};
6. the current mode that the node perceives channel c to be in: `Election` mode, `Converging` mode, or `Converged` mode (see Figure 5.3).

Each node can independently establish the information of parts 3 and 4 by listening on channel c. The information for part 5 is obtained when the SYNC node in channel c listens to the beacon packet from the SYNC node in channel $c+1$. Alternatively, this information can also be obtained when DESYNC nodes in channel c listen for an acknowledgement of a swap request and overhear a DESYNC beacon in channel $c+1$. Finally, the information in part 6 is acquired as described in the following two subsections.

5.2.3.2 `Election` Mode

Election of a SYNC node is initiated in each channel c when N_e consecutive periods have passed without receiving a SYNC beacon packet, or when nodes observe that all other nodes report

the SYNC node ID as NULL. The value of N_e can be set high enough to avoid reelecting a SYNC node just because SYNC beacon packets were lost due to interference. The experiments of Section 5.5 found that $N_e = 10$ provided for virtually no reelections when a SYNC node is already present in each channel c, while allowing for fast network response when a SYNC node actually leaves the channel.

Once the nodes in channel $c \in \{1, \ldots, C\}$ go to `Election` mode, they report this in part 6 of their beacon packets. Each node then randomly generates an 8-bit number, $r \in [0, 255]$, and transmits it in part 3 of its beacon packet. After one complete period, the node with the highest number is elected to become the SYNC node for this channel. In the unlikely case where the highest number is sent by more than one node, the node with the highest node ID (part 2) is elected. All nodes confirm the selection in the subsequent periods by setting their SYNC node ID (i.e., part 3) to the node ID they have just elected. Because beacon packets may occasionally be lost, there may be some sporadic cases where nodes may not unanimously agree to the same elected SYNC node. In such cases, nodes rectify their election according to the majority decision. Once all nodes set the SYNC ID field to the same value, the `Election` mode (i.e., part 6 in the beacon message) changes to either `Converging` or `Converged` mode. This process ensures that (up to) one SYNC node is present per channel. Algorithm 5.1 outlines the SYNC node election process.

Algorithm 5.1 Election Mode

```
1:  if on reception of a beacon packet from SYNC node in channel c then
2:      reset N_e to 0
3:  end if
4:  if on reception of a beacon packet, SYNC is undefined or unheard in channel c then
5:      increment count of missing SYNC node by 1
6:  end if
7:  if more than N_e periods have occured without a SYNC in channel c then
8:      put channel c into Election mode
9:  end if
10: if channel c is in Election mode then
11:     vote uniform(0,255)
12:     process local vote
13: end if
14: for each received vote do
15:     if vote larger than previous maximum then
16:         maintain ID of node and largest vote
17:     else
18:         discard vote
19:     end if
20: end for
21: if local node cast largest vote then
22:     local node enter SYNC mode
23: else
24:     update SYNC node in beacon to highest voting node
25: end if
```

In a real world deployment, the election algorithm may be advantageously modified to reflect nodes energy availability or higher degree of network connectivity, since nodes achieving higher energy harvesting may choose to take the more energy intensive role of becoming a SYNC node.

5.2.3.3 `Converging` Mode via Node Balancing across all C Channels

When nodes are in the `Converging` mode, no channel swapping takes place. However, in order to balance nodes within the available C channels, SYNC nodes can decide to switch to the next channel if fewer nodes are present therein, as described next.

During the `Converging` mode, all nodes apply the DESYNC and SYNC processes of Section 5.2.2. The SYNC node in channel c listens to the next channel for SYNC and DESYNC beacons. By listening to the former, the SYNC node applies phase updates to converge to the synchronous state. By listening to the latter, it establishes the number of nodes present in the next channel, i.e., W_{c+1} (part 5 of beacon packet contents). If

$$W_c - W_{c+1} - 1 \geq 0 \tag{5.4}$$

and $c < C$, then the SYNC node of channel c switches to channel $c+1$ and joins as DESYNC node, thereby triggering a new SYNC node election in channel c (after N_e periods). Importantly, the SYNC node in the highest channel, C, can switch to channel 1, i.e., perform "cyclic" switching from highest to lowest channel, if

$$W_C - W_1 - 2 \geq 0. \tag{5.5}$$

This difference in the switching control for channel C prevents a race condition where nodes would be constantly switching between channels.

Through the new SYNC node for channel c, all nodes remaining in channel c will observe that W_{c+1} increased by one. Furthermore, after N_c consecutive misses of the beacon of the node ID that switched, W_c is decreased by one, i.e., the node is confirmed as having departed channel c. The requirement of N_c consecutive misses before assuming that the node has left channel c avoids erroneously decreasing W_c due to a burst of interference in channel c.

The above process will lead to nodes moving from lower to higher channels, thereby enabling the network to converge to a balanced number of nodes across all C channels. That is, after balancing, there existing $W_c = \{\lfloor \frac{W}{C} \rfloor, \lceil \frac{W}{C} \rceil\}$ nodes in each channel $c \in \{1, \ldots, C\}$. Examples of balancing are illustrated in Figure 5.1(a) and Figure 5.4. The example in Figure 5.4 demonstrates that: *(i)* without the special condition for channel C, SYNC nodes would be cyclically switching in perpetuity; *(ii)* because switching occurs between successive channels during the `Converging` mode, channels with equal numbers of nodes are clustered together,

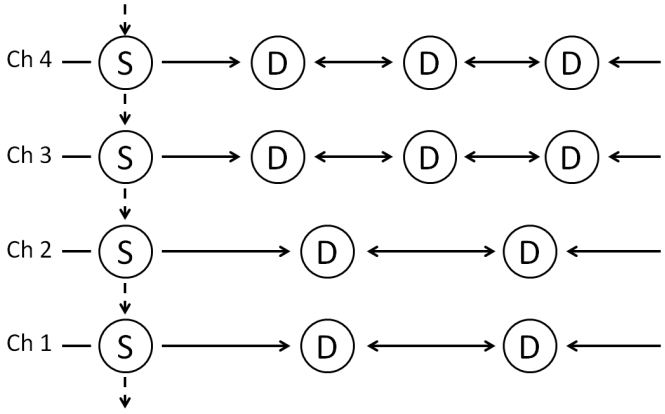


Figure 5.4: Example of balancing under DT-SCS for a network of $W = 14$ nodes in $C = 4$ channels.

with an ascending number of nodes per channel. This is an important feature of the algorithm, as it does not permit channels with unequal node counts to be interspersed. As nodes can only swap with their counterpart in neighbouring channels[5] (a node in channel c may only swap with the concurrently beaconing node in channel $c+1$), ensuring that channels with equal numbers of nodes are grouped together greatly improves connectivity in the network. Algorithm 5.2 outlines the process of channel balancing.

Algorithm 5.2 Channel Balancing

1: **for** SYNC beacon packet received in channel $c+1$ **do**
2: **if** fewer nodes in channel $c+1$ than in channel c (difference of two $c = C+1$) **then**
3: move to the next channel, wrapping at $C+1 \rightarrow 1$
4: **end if**
5: **end for**

5.2.3.4 `Converged` Mode, Channel Swapping and Data Transmission

Once nodes are in `Converging` mode and their SYNC or DESYNC beacon packets fall within the convergence threshold, i.e., (5.2) holds, they switch to the `Converged` mode. Nodes can thus begin data transmission following a short guard time interval after their beacon packet broadcast. The duration of their transmission lasts until another short guard time interval prior to the subsequent node beacon packet broadcast, as shown in Figure 5.1(b). In `Converged` mode, nodes transmit data, send or acknowledge swap requests and swap channels.

Limited listening: Beyond the time required for receiving (or transmitting) data packets, all nodes switch on their transceiver only during beacon guard times [Figure 5.1(b)]. This limits the required listening to short time intervals within each period of T sec. The guard time is used to allow for beacon variability due to SYNC or DESYNC beacon time adaptation via phase

[5] In the network configuration of Figure 5.4, DESYNC nodes can apply channel swapping only between channel 1 and channel 2, and between channel 3 and channel 4. However, SYNC nodes can still swap places between all four channels, as their beacon packet transmissions remain synchronous.

updates. This adaptation may cause beacon time fluctuations, the range of which is controlled via the coupling parameters α and β. To reduce the listening time further, all nodes can switch to *sparse listening*, that is, they can opt to listen for beacons only once every several periods.

Channel Swapping: In `Converged` mode all DESYNC nodes of each channel c can opt to transmit swap requests in the next channel (or acknowledge swap requests from a node of channel $c-1$) if, and only if, $W_{c+1} = W_c$ (or $W_c = W_{c-1}$), with $W_{C+1} \equiv W_1$ and $W_0 \equiv W_C$. If a swap is acknowledged, the corresponding SYNC or DESYNC nodes swap channels in their subsequent beaconing cycle and remain in the new channel until another swap RQ/ACK event. Because the swap acknowledgement may not always be received by the requesting node, sporadic cases may occur where the node requesting the swap does not actually swap channels. To overcome this, every node that received a swap request transmits its first beacon packet towards the end of the guard time after performing the channel swap. This enables the node to detect that its swap-requesting "partner" is not sending its beacon in its old channel and has indeed carried out the swap. If, however, the swap partner did not carry out the swap, then the node returns to its original channel, resumes beaconing therein and requests a new swap. Algorithm 5.3 outlines this process.

Algorithm 5.3 Channel Swapping

```
1:  if swap request received from c − 1 or c + 1 then
2:      send acknowledgement to swap request
3:      send an early beacon packet in new channel
4:      if requesting node does not beacon in original channel c − 1 or c + 1 then
5:          swap was successful
6:      else
7:          swap failed
8:          return to previous channel
9:      end if
10: end if
```

By using the channel swapping mechanism, DT-SCS ensures each node can attempt to swap channels whenever: *(i)* the application requires; *(ii)* a node must reach nodes not present in its current channel; *(iii)* excessive interference is observed in a channel. Channel swapping should not be confused with channel switching: the former is done in a peer-to-peer RQ/ACK manner in `Converged` mode, while the latter is performed in order to balance the total nodes in C channels during the `Converging` mode, and it does not use an RQ/ACK mechanism. Finally, as depicted in Figure 5.3, nodes in the `Converged` mode may move back to `Converging` mode if N_c consecutive beacon packets are not received from any SYNC or DESYNC node. Nodes move to `Election` mode if N_e consecutive SYNC beacon packets are not received. In both of these modes, no data transmission or channel swapping takes place and nodes listen constantly.

Coupling Adaptation: As mentioned in Section 5.2.1, losing beacon packets due to interfer-

ence may lead to beacon packet transmissions at incorrect times. To absorb transient oscillations of beacon times, while at the same time maintain fast convergence, the values of α, β, N_e, N_c can be adjusted per node. Setting $\alpha, \beta \to 1$ and $N_e, N_c \leq 3$ allows for very quick convergence and better suits channels experiencing low interference. Conversely, setting $\alpha, \beta \to 0$ and $N_e, N_c \geq 8$ provides for more stable operation under interference, albeit at the cost of slower convergence and reaction time. While the joint optimisation of these parameters with DT-SCS deployments remains a future research topic, the reader is referred to existing work on the impact of α and β in single channel DESYNC and SYNC [32, 33].

5.3 Protocol Analysis

5.3.1 Balancing and Stability

As described in Section 5.2.3.3, during the `Converging` mode of the proposed DT-SCS protocol, SYNC nodes can decide to switch to the next channel if they detect fewer nodes present therein. The following proposition proves that this mechanism leads to a balanced number of nodes per channel as illustrated in Figure 5.1(a) and Figure 5.4.

Proposition 5.1. The proposed node balancing mechanism converges to $W_c \in \left\{\lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor\right\}$ nodes within each channel $c \in \{1, ..., C\}$.

Proof: See Appendix 7.2.1.

Once C channels have balanced numbers of nodes, the DT-SCS protocol performs repeated PCO synchronisation [33] across channels and desynchronisation [32] within each channel. The former technique leads to synchronised beacon transmissions of SYNC nodes across channels, while the latter ensures fair TDMA scheduling between the nodes in a channel. The following proves the efficacy of the algorithm.

Proposition 5.2. For each channel c, the proposed DT-SCS protocol converges to equidistant beacon packet transmissions at intervals of $T\left(\frac{1}{W_c} \pm b_{\text{thres}}\right)$ seconds, with $W_c \in \left\{\lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor\right\}$ *and the* SYNC *beacons are broadcast concurrently in all channels.*

Proof: See Appendix 7.2.2.

5.3.2 Connectivity

Via channel swapping, the SYNC node in each channel c can eventually reach any node in the remaining channels $\bar{c} \in \{1, \ldots, C\}$, $\bar{c} \neq c$, except for other SYNC nodes, since they are

concurrently transmitting. Hence, the degree of connectivity of a SYNC node is

$$D_{\text{SYNC}} = W - C. \tag{5.6}$$

Similarly, for all channels with $\lfloor \frac{W}{C} \rfloor$ or $\lceil \frac{W}{C} \rceil$ nodes, all DESYNC nodes can swap channels in order to reach any other SYNC or DESYNC node, except for the DESYNC nodes that are synchronous to them. In the `Converged` mode, the

$$C_{\text{high}} = W - \left\lfloor \frac{W}{C} \right\rfloor C \tag{5.7}$$

highest channels will have

$$W_{\text{DESYNC,high}} = \left\lceil \frac{W}{C} \right\rceil - 1 \tag{5.8}$$

DESYNC nodes (and one SYNC node), while the

$$C_{\text{low}} = C - \left(W - \left\lfloor \frac{W}{C} \right\rfloor C \right) \tag{5.9}$$

lowest channels will have

$$W_{\text{DESYNC,low}} = \left\lfloor \frac{W}{C} \right\rfloor - 1 \tag{5.10}$$

DESYNC nodes (and one SYNC node).

Proposition 5.3. The average degree of connectivity of a DESYNC node is

$$\begin{aligned} D_{DESYNC} = & \frac{1}{W-C}[(C_{high} W_{DESYNC,high})^2 \\ & + (C_{low} W_{DESYNC,low})^2 \\ & + C_{low} C_{high} \times (W_{DESYNC,high} + W_{DESYNC,low})]. \end{aligned} \tag{5.11}$$

Proof: See Appendix 7.2.3.

In the example of Figure 5.1, $D_{\text{SYNC}} = 9$ and $D_{\text{DESYNC}} = 9$, while in the example of Figure 5.4 (and following the node placement of Figure 2.6), $D_{\text{SYNC}} = 9$ and $D_{\text{DESYNC}} = 7.2$. For the same wireless node placement, TSCH achieves average connectivity of 3.5 under its default configuration (see Section 5.5 for details), which is illustrated at the bottom of Figure 2.6.

5.3.3 Estimation of Convergence Time

The protocol initiates with W nodes randomly joining C channels. To estimate the expected time for DT-SCS to converge to the steady state, an estimate of the probability that the ensemble of W nodes will reach combination i out of $\mathcal{C}_{W,C}$ possible combinations, with each

combination comprising $[W_1(i) \dots W_C(i)]$ nodes within C channels is required. The convergence time estimate is summarised in the following proposition.

Proposition 5.4. *Under no packet losses and no hidden nodes in the network, the expected delay until convergence in DT-SCS is*

$$d_{W,C} = TN_e \sum_{i=1}^{\mathcal{C}_{W,C}} \left(\Pr(i) \max_{\forall c} \left| W_c(i) - \left\lfloor \frac{W}{C} \right\rceil \right| \right). \tag{5.12}$$

with

$$\mathcal{C}_{W,C} = \frac{(W+C-1)!}{(C-1)!W!}. \tag{5.13}$$

and

$$\Pr(i) = \prod_{c=1}^{C-1} \left[\binom{W_{\text{res},c}(i)}{W_c(i)} \frac{(c-1)^{W_{\text{res},c}(i)-W_c(i)}}{c^{W_{\text{res},c}(i)}} \right]. \tag{5.14}$$

with $\forall i : W_{\text{res},c}(i) = W - \sum_{m=1}^{c-1} W_m(i)$.

Proof: See Appendix 7.2.4.

5.3.4 Estimation of Energy Consumption

It is generally accepted that the radio chipset draws the most power in low-power ad hoc wireless network deployments [21, 28]. As an illustration, Table 2.2 on page 28 presents the power requirements for different energy states of TelosB at 3 Volts with the CC2420 transceiver [29, 20].

The energy consumption of a single node is broken down into two parts, receiving and transmitting. The energy use due to receiving is analysed for `Converging` and `Converged` modes. While the channel is in `Converging` mode, nodes listen continuously. Once converged, nodes listen to the data slots of other nodes and to swap requests. These parameters are summed to give the total amount of energy dissipate to receive by each DT-SCS node (SYNC or DESYNC), i.e.,

$$E_{\text{Rx}} = P_{\text{Rx}} \left(d_{\text{conv}} + n_{\text{SS}} \times T \times \frac{W'_c}{W_c} \right) \tag{5.15}$$

where P_{Rx} is the power of the transceiver when receiving, d_{conv} is the time required for DT-SCS to converge as estimated by Proposition 5.4, n_{SS} is the number of periods the network operates in `Converged` (SS) mode, and W'_c is the number of nodes the node listens to. Likewise, the energy dissipated for transmission is split into the energy used to *(i)* broadcast beacon packets and swap RQ/ACK and *(ii)* to transmit data. Assuming that each node broadcasts on average one beacon packet and one swap request or acknowledgement per period T, and that each node

transmits data for every interval of $\frac{T}{W_c}$ seconds following its beacon packet, the total amount of energy dissipated for transmission is

$$E_{\text{Tx}} = P_{\text{Tx}} \left[2 \times t_{beacon} \frac{d_{\text{conv}}}{T} + n_{\text{SS}} \times \frac{T}{W_c} \right] \tag{5.16}$$

with P_{Tx} the transmit power, t_{beacon} the time taken to transmit a beacon or swap RQ/ACK message and $\frac{d_{\text{conv}}}{T}$ the expected number of beacon transmissions until convergence, where converge time, d_{conv} , is estimated by Proposition 5.4.

5.4 Simulation Results

All simulations for DT-SCS were performed in MATLAB, by extending the event driven simulator for the DESYNC protocol by Degesys *et al.* [32]. Since DT-SCS is a MAC layer protocol, the simulation reports results in function of packet loss experienced within each of the 16 channels of the PHY. The results are compared against TSCH simulation results produced via the 6tisch simulator [2], which is the most accurate TSCH simulator available in the public domain. Since the 6tisch simulator allows for link and timeslot establishment between nodes, it was extensively modified to also simulate the operation of the EM-MAC protocol [3], which is one of the most prominent decentralised protocols in the literature. Specifically, the use of the control channel for slot RQ/ACK is disabled and instead each node: *(i)* joins the network by selecting channels and wake-up times pseudo-randomly; *(ii)* sends beacon packets notifying senders about their listening slots and wake-up times; *(iii)* predicts the wake-up times and channels of receiver(s) and join them to send packets as per the established configuration; *(iv)* blacklists channels if packet loss above 15% is observed, as per the original EM-MAC proposal [3]. Overall, this comparison is indicative for a broad range of wireless networks encountered in vehicular or mobile node environments because other solutions, like the IEEE 1609.4 extension [94, 78] of the IEEE 802.11p [23] standard, also use slotframe and reservation mechanisms similar to TSCH.

Here, the data payload size of 60 bytes. Packet loss is simulated by randomly dropping packets to mimic interference conditions experienced within the 2.4 GHz unlicensed band. Simulations were repeated 100 times and average results are reported. In the vast majority of the reported results, the span of 95% confidence intervals was found to be only ±15% from the average values.

Unless otherwise stated, all simulations assume an ad hoc network consisting of $W = 64$ nodes in the $C = 16$ channels of IEEE 802.15.4. Concerning the configuration of the proposed DT-SCS, $\alpha = 0.6$ and $\beta = 0.6$ were used for the DESYNC and SYNC parameters of (5.1) and

(5.3), with $T = 100$ ms, $t_{\text{swap}} = 12$ ms[6], $b_{\text{thres}} = 0.01$ and $N_{\text{e}} = N_{\text{c}} = 10$. Under the specified settings and excluding the guard time periods (13 ms), one data packet of 60 bytes can be sent between two consecutive beacon packets within the same channel.

Regarding TSCH, the default 6tisch settings for timeslots (101 slots) per slotframe and channels (16 channels) were used. Each node has, on average, two outgoing (data sending) links and one incoming (data receiving) link. In addition, the `--traffic` parameter of 6tisch is set to 0.75, which, under the established setup, corresponds to two timeslots per node link within each slotframe. Convergence is assumed for TSCH when ten consecutive slotframes are observed with less than 5% change in timeslot allocations amongst nodes.

Finally, concerning EM-MAC, following the low duty cycle of the original paper [3], one outgoing and one incoming slot per node is used within the 16 channels available, with maximum sleep time interval per node equivalent to 100 slots. The wakeup slot duration was set to be equal to the slot duration of TSCH. The use of these settings ensured minimal clock drift between transmitters and receivers in EM-MAC, and therefore the exponential chase algorithm proposed in the original implementation of the protocol was found to be unnecessary in this implementation. Convergence is assumed for EM-MAC when at least 70% of the nodes have established the wake up time pattern (and channel) to send to their receiver node. The use of 70% was found to provide for the best compromise between convergence and robustness to packet loss and clock drift between sender and receiver nodes.

5.4.1 Node Balancing and Connectivity

The first set of simulations show that the proposed node balancing mechanism within DT-SCS converges to $\lfloor \frac{W}{C} \rfloor$ or $\lceil \frac{W}{C} \rceil$ nodes per channel. Figures 5.5(a)-(b) show the initial and final node beacon packet phases versus the channel number for $W = 14$ nodes in $C = 4$ channels. In the initial state [see Figure 5.5(a)], a random number of nodes, each with a random phase, enter each channel. In this example, the initial state is $W_{\{1,2,3,4\}} = [5, 3, 2, 4]$. In the converged state, [see Figure 5.5(b)], the nodes have been balanced within the channels (with the elected SYNC nodes indicated in red), where the two highest channels have four nodes and the two lowest channels have three nodes.

5.4.2 Convergence Time

An important aspect of the proposed protocol is the time required to reach the `Converged` mode from a random initial state. Table 5.2 presents the average convergence time of DT-SCS

[6] The reported values of T and t_{swap} were chosen such that, under the expected number of nodes per channel in steady-state (i.e., 4 nodes), the duration of the data payload interval in-between the guard times becomes 13 ms, which is similar to the data payload interval of TSCH.

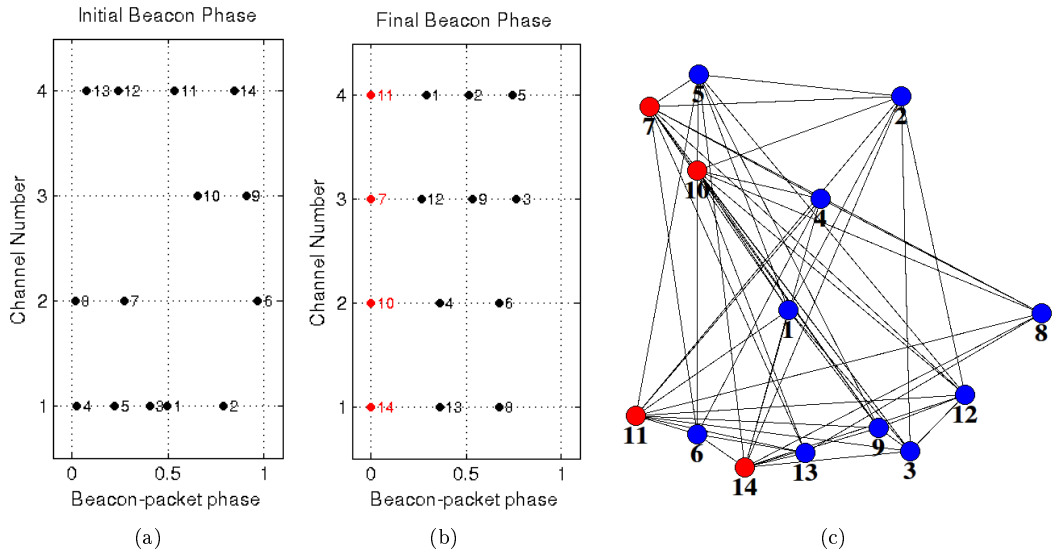


Figure 5.5: Initial (a) and final (b) node beacon packet phase locations versus channel number. Each node has a unique ID, with SYNC nodes indicated in red. (c) Corresponding connectivity between DT-SCS nodes in the `Converged` mode, with node swapping enabled.

Table 5.2: Theoretical (Proposition 5.4) vs. simulation convergence time of DT-SCS under various settings and no packet losses.

$\{W, C\}$	Simulation	Proposition 5.4	% Error
$\{64, 16\}$	1.296	1.123	13.4
$\{48, 12\}$	1.138	1.064	6.5
$\{32, 4\}$	1.3850	1.410	1.8
$\{25, 3\}$	0.958	1.013	5.7
$\{12, 3\}$	0.509	0.662	30.1
$\{8, 2\}$	0.308	0.419	35.1

versus the corresponding theoretical result of Proposition 5.4 under a variety of settings for W and C (all other settings are left as described previously) and no packet loss. Evidently, for the majority of cases, Proposition 5.4 predicts the simulation convergence time with less than 15% error, and the maximum prediction error is below 36%. Importantly, under no packet loss, the DT-SCS convergence time was always found to be below 1.5 s for all settings.

Next, the convergence of DT-SCS is investigated under the occurrence of packet losses, and also in comparison to the time required by TSCH and EM-MAC to achieve a stable contention free slot allocation via their centralised and distributed advertising mechanisms. Figure 5.6 presents the related results for $W = 64$ nodes under varying packet loss percentage imposed on each of the 16 channels of IEEE 802.15.4. Even though these simulations do not incorporate all the aspects of propagation and interference experienced in a real testbed, the results in Figure 5.6 demonstrate that DT-SCS reduces the required convergence time by 22.04–91.61% in comparison to TSCH. Such quick convergence occurs because, contrary to TSCH, the proposed DT-SCS protocol does not require nodes to advertise and acknowledge free slots, which is

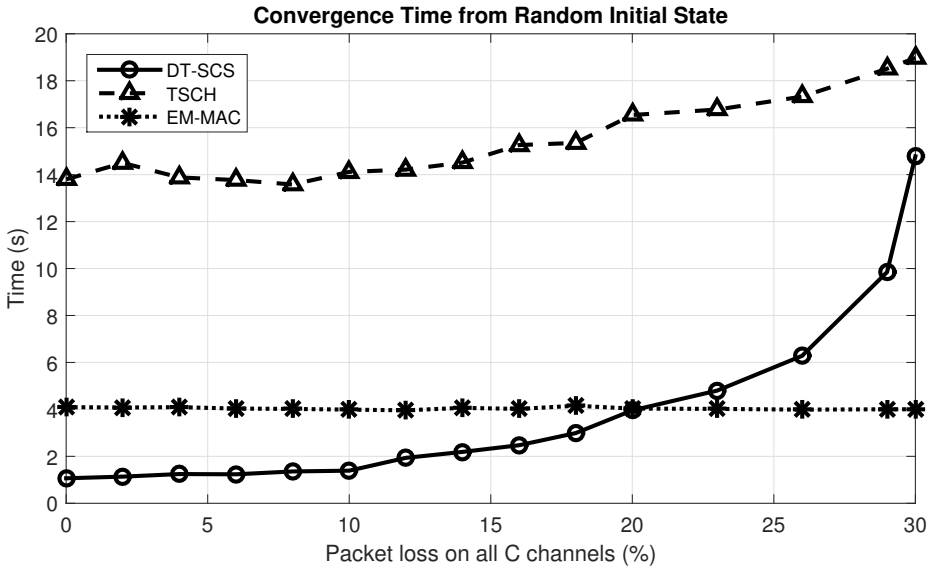


Figure 5.6: Average time required for convergence in DT-SCS, TSCH and EM-MAC when 64 nodes join 16 channels randomly during initialisation.

a process that is detrimental to the convergence time. In addition, DT-SCS converges to the steady-state faster than EM-MAC for packet loss rates below 20%. However, EM-MAC exhibits a very stable convergence behaviour as the low duty cycle and low connectivity of the protocol ensure that, even under high packet loss rate, the majority of nodes establish the wake-up pattern to rendezvous with their receiver within 4 seconds.

Subsequently, the time required for the network to return to the steady state under the effect of churn (which is typically encountered in mobile and vehicular networks) is studied. In this case, the simulation is started from an initially in steady-state, but the arrival or departure of nodes (i.e., the effect of churn) cause the network to return to `Converging` mode. Figure 5.7 depicts a comparison of the re-convergence speed of the proposed DT-SCS against that of TSCH for different churn conditions; low, medium and high churn. These conditions correspond to 5%, 25%, and 50% of nodes arriving or leaving the network, respectively. The results show that, under medium and high churn and packet loss rates up to 22–25%, DT-SCS reduces the time that the network requires to return to steady-state in comparison to TSCH and EM-MAC (where only the average results are shown as all churn cases exhibited very similar behaviour). This is because under medium and high churn all protocols will require extensive reconfiguration to return to steady-state. Similar to the convergence from a random initial state, the proposed DT-SCS achieves quicker convergence in comparison to TSCH. On the contrary, TSCH and EM-MAC offer faster convergence when the packet loss rate is high, or when low churn is experienced. This is to be expected since, under low churn, only the TSCH or EM-MAC nodes that have lost communication links will engage in re-advertising actions. Conversely, the

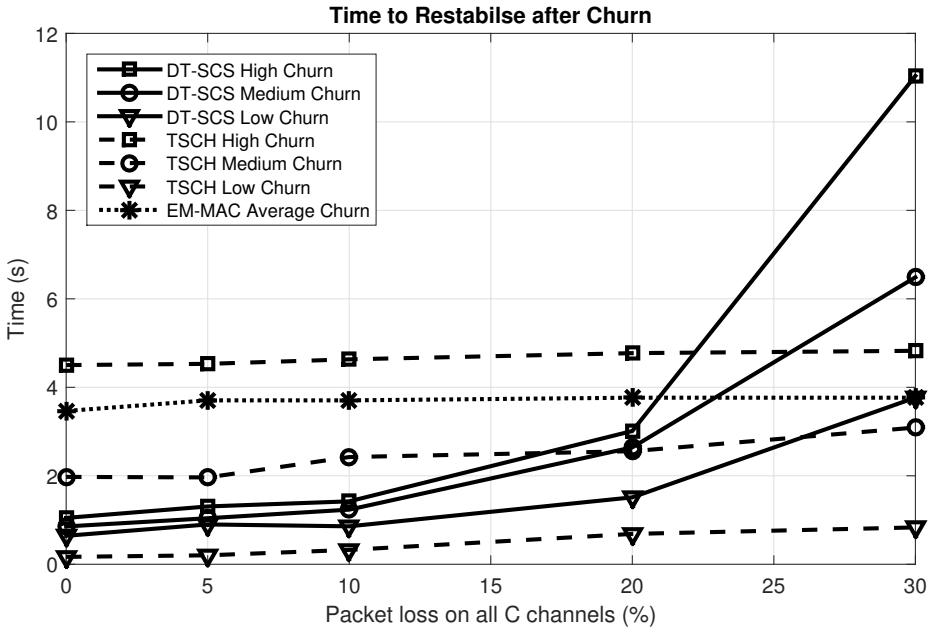


Figure 5.7: Average time required for DT-SCS, TSCH and EM-MAC to return to steady-state under varying degrees of node churn.

proposed DT-SCS protocol will force all nodes to re-converge. Moreover, under high packet loss, few nodes receive advertising RQ/ACK, and so the schedule remains largely stable. In both cases, however, the disadvantage is that not all abandoned TSCH slots (or EM-MAC sender-receiver pairings) are reoccupied, thereby leading to lower bandwidth utilisation.

Next, an investigation into the convergence speed of DT-SCS when some of the nodes in the network are hidden from other nodes is undertaken. In particular, the time to achieve convergence to steady-state when a number of randomly chosen nodes (both SYNC and DESYNC included) cannot communicate with a random subset of twelve other nodes in the considered setup is measured. In order to make an extensive investigation of the effect of hidden nodes, the number of nodes is varied from 0 to 32 and the DT-SCS convergence process repeated multiple times in order to measure the average convergence time. The results in Figure 5.8 show that, irrespective of the presence of hidden nodes, the convergence speed of DT-SCS is significantly higher than that of TSCH and EM-MAC. When hidden nodes are present, the required convergence time of DT-SCS increases by up to 3 seconds, while that of TSCH decreases by up to 3 seconds (albeit still remaining almost three times higher than that of DT-SCS). This is to be expected since TSCH nodes simply miss RQ packets from hidden nodes. On the other hand, the `Converging` mode of DT-SCS will perform channel switching until all nodes join channels with non-hidden terminals.

The convergence time of the proposed protocol against TSCH under the effect of targeted interference is also studied, i.e., high packet losses on a given channel. In this regard, following

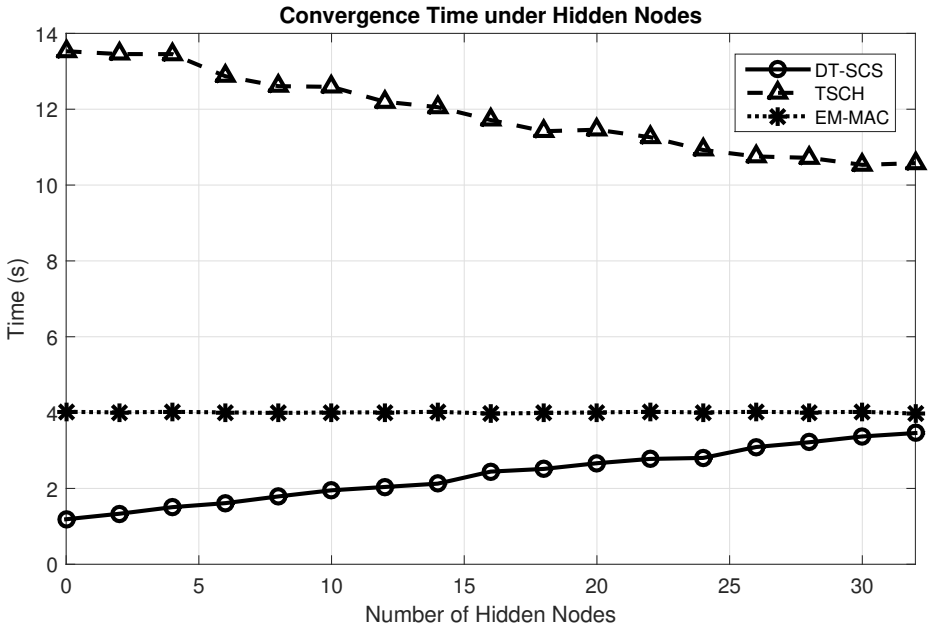


Figure 5.8: Average convergence time under increasing number of hidden nodes.

Table 5.3: Average Convergence Time (in seconds) under Targeted Interference.

	DT-SCS	TSCH	EM-MAC
On a random channel ($c \neq 1$)	1.2496	14.2186	4.095
On TSCH control channel	1.2496	73.9126	4.095

experiment was devised: Packet loss of 30% on is applied to channel $\hat{c}$ of DT-SCS, TSCH and EM-MAC, while all other channels $c \neq \hat{c}$ suffer from packet loss of 2%. Two cases are explored: *(i)* when $\hat{c}$ is a random channel ($\hat{c} \in \{1, \ldots, 16\}$), or *(ii)* $\hat{c}$ is the control channel of TSCH and a specific channel (e.g., $c = 1$) of DT-SCS or EM-MAC. Table 5.3 shows that the convergence time of all protocols is increased with targeted interference. However, contrary to the proposed DT-SCS and EM-MAC, TSCH is particularly vulnerable to packet losses on the control channel, whereby the convergence time is increased by 444%. This underlines the importance of the decentralised, infrastructureless, nature of the proposed protocol and EM-MAC, and highlights potential problems with centralised protocols that rely on control nodes or coordination channels. Furthermore, under high control channel interference, a network deployment using TSCH would struggle to maintain time synchronisation across all channels, thereby suffer from a further loss of performance.

5.4.3 Bandwidth Efficiency

To assess the steady state performance of the proposed DT-SCS against TSCH and EM-MAC, the total payload bits successfully received are measured on all DT-SCS nodes per second versus the equivalent results obtained via the 6tisch simulator for TSCH (and its modification

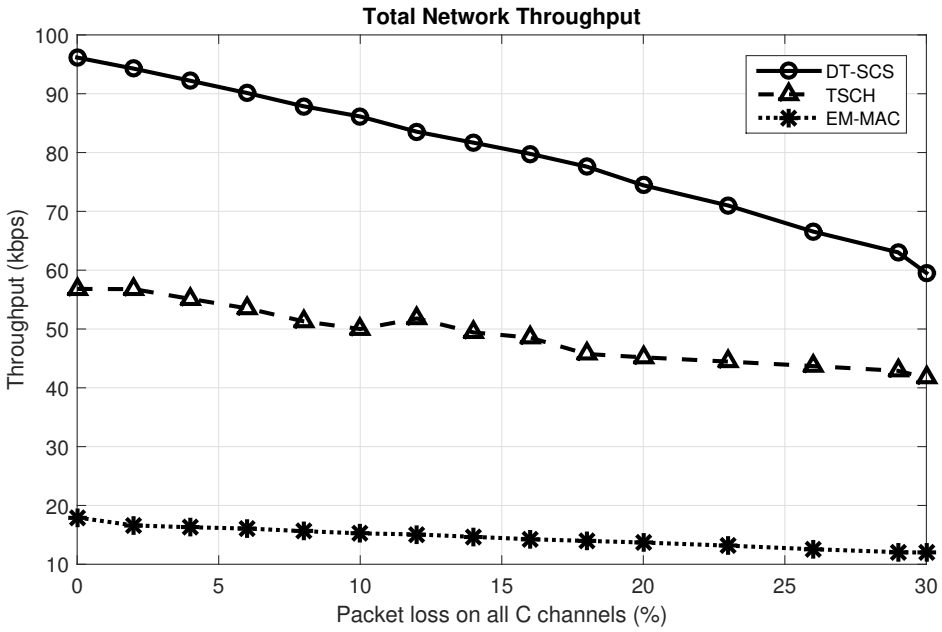


Figure 5.9: Comparison of bandwidth utilisation (total payload transmitted by all nodes per second) between the proposed DT-SCS, TSCH and EM-MAC.

for the EM-MAC simulation). Figure 5.9 shows that the DT-SCS protocol approach achieves a substantially higher slot and channel utilisation than TSCH, leading to bandwidth gains of 27.12–40.63%. At the same time, it offers more than five times the network throughput of EM-MAC. This is because DT-SCS allows for all nodes to use all available time in between their own beacon and the next beacon (barring the guard time intervals) for contention free transmission. On the contrary, TSCH requires advertisement and confirmation actions and imposes a rigid slot allocation. Such a rigid slotframe allocation imposes strict limitations on the available bandwidth per node, restrictions that are not applied by the proposed DT-SCS protocol. Moreover, EM-MAC imposes a low duty cycle due to the receiver driven rendezvous policy applied in the protocol, thereby leading to substantially lower network throughput.

5.5 Experiments With TelosB Motes

Since the simulation experiments of Section 5.4 showed that EM-MAC achieves substantially lower network throughput and lower connectivity than the proposed DT-SCS, EM-MAC was not considered in the hardware experiments. The remaining proposed DT-SCS and TSCH were implemented as applications in the Contiki-OS 2.7 IoT operating system running on low-power TelosB motes. By using the NullMAC and NullRDC network stack options in Contiki-OS, all node interactions at the MAC layer can be controlled via application layer code. The code was developed directly on the mote hardware with the Cooja Simulator (part of Contiki-OS)

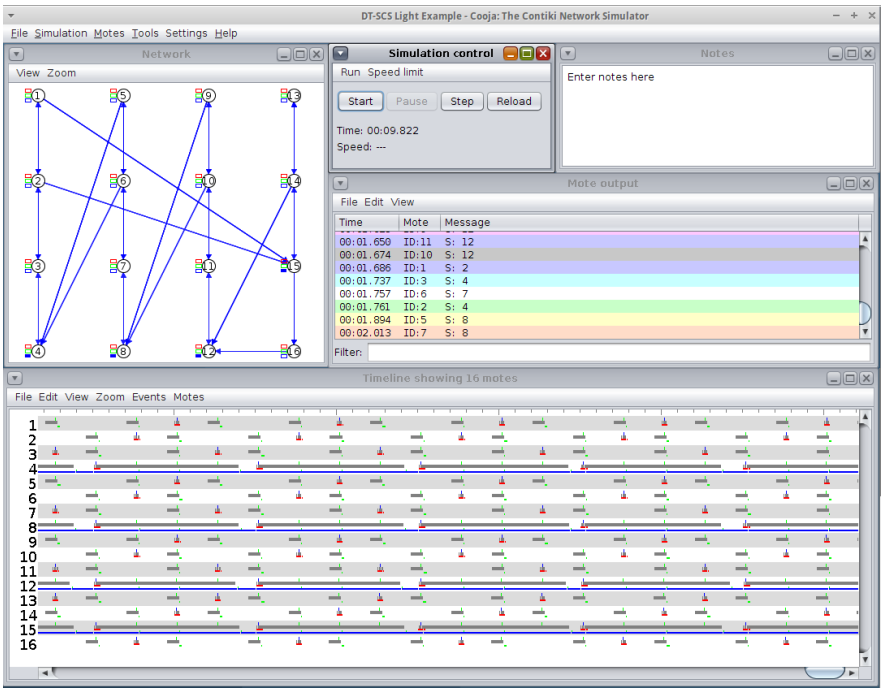


Figure 5.10: Cooja is a network simulator bundled with Contiki-OS, which allows networks of motes to be simulated. Here, the TelosB motes are simulated at hardware level. The network topology is visible in the top left of the image. The bottom shows a timeline of each node's activity which clearly shows node alignment and spacing within channels, as well as node radio status during *limited listening*.

allowing for more in-depth debugging and understanding, which is preferable to the time variability incurred by outputting debugging information via a node's serial port. The DT-SCS protocol application code for Contiki-OS 2.7 can be downloaded from the experiment webpage, `https://github.com/m1geo/DTSCS`, and is also included for completeness in Appendix 7.3.2. Figure 5.10 shows the proposed DT-SCS protocol running in the steady state within the Cooja. The hardware DT-SCS implementation follows the protocol description of Section 5.2.3 with $T = 100$ ms, $\alpha = \beta = 0.6$, $N_e = N_c = 10$ and guard time of 12 ms [Figure 5.1(b)] for increased robustness to interference. During the `Converged` mode all nodes switch to "sparse listening", i.e., they listen for beacons only once every twenty periods, unless high interference noise is detected[7]. Concerning TSCH, the hardware implementation follows the advertisement RQ/ACK and slotframe structure of the 6tisch simulator and TSCH standard [81, 28, 2]: Channel 11 of IEEE 802.15.4 PHY is used for advertisements, RQ/ACK ratio of 1/9, slotframe comprising 101 slots of 15ms each, and one node (at the centre of the deployment) was set to broadcast the slotframe beacon for global time synchronisation. For both frameworks, the TelosB high

[7] In the `Converged` mode, the interference noise floor is determined in between transmissions by reading the CC2420 received signal strength indicator (RSSI) register and switch to regular listening of all anticipated neighbouring beacons per period if high interference noise levels are detected. Therefore, the option of sparse listening does not affect the stability of DT-SCS.

Figure 5.11: Example of one of the four rooms comprising an experimental setup. The four right-most nodes in the image are used for noninvasive network observation, while the RF signal generator, which acts to generate interference, is shown in the background.

resolution timer (`rtimer` library) was used for setting transmission and listening events[8]. A remark that the slotframe period and guard time settings are similar to those of the IEEE 1609.4 [94, 78] extension of IEEE 802.11p [23].

Similar to the above simulations, all experiments are based on the deployment of $W = 64$ nodes in the $C = 16$ channels of IEEE 802.15.4 PHY. For DT-SCS, this leads to $W_c = 4$ nodes per channel in the steady state. The 64 TelosB motes were placed in four neighbouring rooms on the same floor of an office building, with each room containing 16 nodes (plus an additional 4 passive [noninvasive] monitoring nodes). Figure 5.11 shows an example of one such room, with three remaining rooms located at a larger distance, each encompassing a further 16 nodes. Each DT-SCS node (either SYNC or DESYNC) could reach up to 48 other nodes via channel swapping. This agrees with the values for D_{SYNC} and D_{DESYNC}. On the other hand, each TSCH node could reach only up to four other nodes under this configuration. Overall, the setup corresponds to scenarios involving dense network topologies and data intensive communications once the ad hoc wireless nodes are activated from a suspended state.

5.5.1 Power Dissipation

The average power dissipation of DT-SCS and TSCH nodes was measured by placing selected TelosB motes in series with a high-tolerance 1-Ohm resistor and utilising a high-frequency

[8] The `rtimer` library requires a slight modification to allow multiple updates before the callback has expired. See Appendix 7.3.2.2 for more information and a software patch file

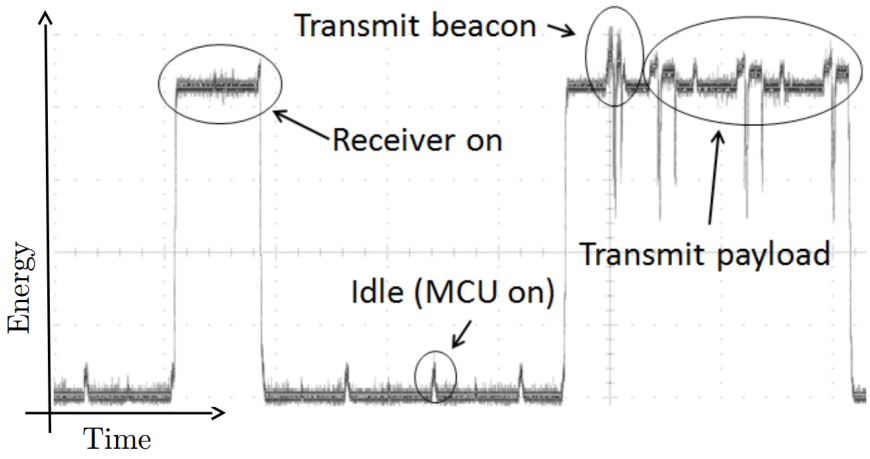


Figure 5.12: Oscilloscope snapshot depicting the instantaneous energy consumption of a TelosB mote under the proposed DT-SCS. When no payload is transmitted, energy is consumed by the processor (MCU) and the radio chipset that transmits and listens for beacons.

Tektronix MDO4104-6 oscilloscope to capture the current flow through the resistor in real time. For the power dissipation experiment, no other devices (or interference signal generators) operating in the 2.4 GHz band were present in the surrounding area. Average results collected over five minutes of operation are reported. A snapshot of the oscilloscope showing the power consumption profile of a TelosB mote using DT-SCS in `Converged` mode is given in Figure 5.12. The average power dissipation of DT-SCS for notes to maintain network operation without transmitting or receiving payload data was measured to be 1.62 mW for DESYNC nodes and 2.08 mW for SYNC nodes.

The theoretically expected value, [estimated via (5.15) and (5.16) with power values corresponding to the transmit, receive and sleep mode of the CC2420 transceiver] was found to be 1.31 mW. This validates the implementation against the theoretical analysis.

The average power dissipation of a TSCH node under minimal payload (128 bytes per 4 s) was found to be 1.64 mW, which is very close to the value that has been independently reported by Vilajosana *et al.* [28]. Therefore, under the same setup, DT-SCS and TSCH were found to incur comparable power dissipation for their operation.

5.5.2 Results under Interference

The convergence time of DT-SCS and TSCH under varying interference levels is investigated. Rapid convergence to the steady network state is extremely important when the entire set of nodes is initiated from a suspended state, or when sudden changes happen in the network (e.g., multiple nodes join or leave). A test was carried out with 100 independent runs, with each room containing an interference generator for 25 tests. To generate interference, an RF signal generator was used to create an unmodulated carrier in the centre of each channel. The carrier

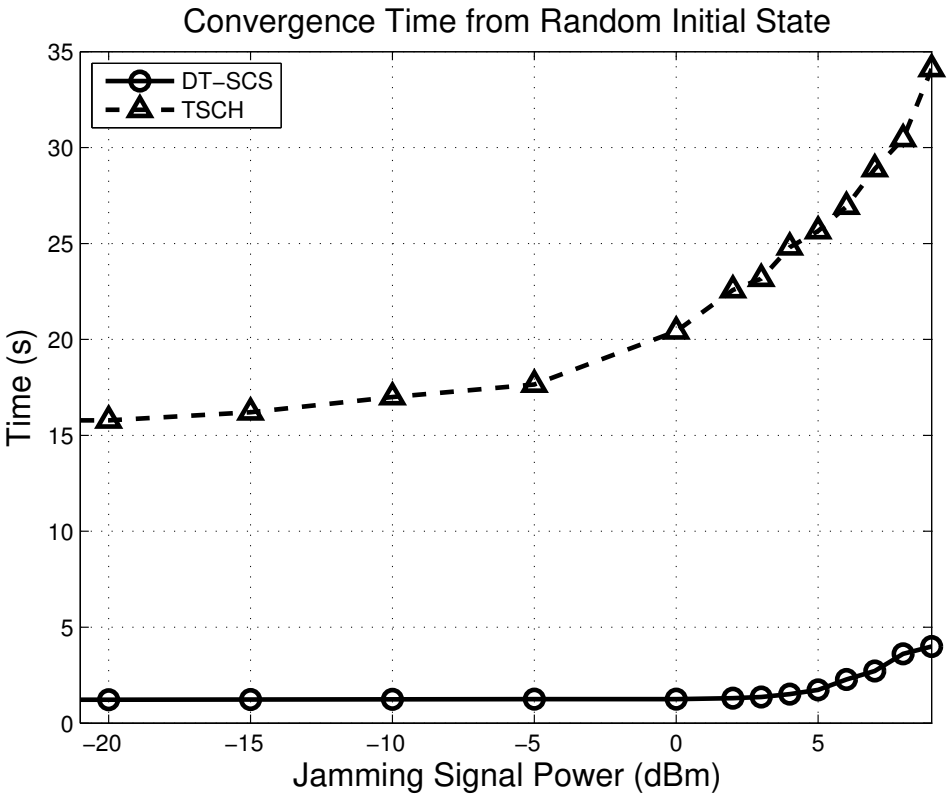


Figure 5.13: Average time required for DT-SCS to reach `Converged` mode and for TSCH to reach a stable slotframe allocation under varying interfering signal power levels.

amplitude was adjusted to alter the signal-to-noise-ratio (SNR) at each receiver [147]. The nodes were set to maximum transmit power (+0 dBm) in order to operate under the best SNR possible. As an indication, in the utilised experimental environment, jamming signal powers of 6.00, 9.00, 10.00 and 10.25 dBm correspond to average packet losses of 0.3%, 1.7%, 11.4%, 29.4%, respectively.

Figure 5.13 shows the time required for DT-SCS and TSCH to converge under varying interfering signal power levels. The results obtained from the hardware implementation corroborate that the proposed DT-SCS reduces the convergence time by an order of magnitude in comparison to TSCH. Moreover, the difference in convergence time between the proposed protocol and TSCH is increasing with the interference level. This result demonstrates the key advantages of the DT-SCS protocol with respect to TSCH: *(i)* it is fully decentralised and *(ii)* it does not depend on an advertisement and acknowledgement scheme.

Next, an investigation into the convergence time of the proposed DT-SCS protocol and TSCH under the effect of targeted interference on a given channel is conducted. Concerning DT-SCS, given that there is no coordination channel, an exploration on how the interference in channel $c+1$ effects the convergence in channel c is instead carried out. A moderate level of interference (that is, +5 dBm) in channel $c+1$ causes fluctuations in the SYNC node beacon

of channel $c \in \{1, \ldots, 16\}$, which in turn causes the average convergence time to increase from 1.223 to 1.518 seconds. When the same level of interference is also applied on channel c , the convergence time is further increased to 2.738 seconds. Regarding TSCH, interference in the advertisement channel led to unstable behaviour and, for the cases where convergence was eventually achieved, more than 30 seconds were required. This demonstrates the detriment of depending on a coordination channel for advertisements.

5.5.3 Bandwidth Results

The total network bandwidth (that is, total payload bits per second) achieved under DT-SCS and TSCH was measured. Interference was applied as described in Section 5.5.2. The results, depicted in Figure 5.14, show that DT-SCS systematically achieves more than a 40% increase in the total network throughput, irrespective of the interference level. Both protocols experience a significant loss of throughput under high interference levels (that is, above +10 dBm), which is, however, substantially more severe for TSCH. In effect, when interference is above +12 dBm, the bandwidth obtained with TSCH drops to zero because of the inability of TSCH to recover lost slots through advertising. On the contrary, even at high interference levels, DT-SCS recuperates bandwidth utilisation due to the elasticity of SYNC and DESYNC mechanisms and the high values used for N_e and N_c.

5.6 Conclusions

A novel protocol for ad hoc wireless networks was proposed, that performs decentralised time-synchronised channel swapping (DT-SCS) and circumvents certain convergence and network utilisation problems of existing designs, such as the state-of-the-art time-synchronised channel hopping (TSCH) protocol. The unique aspect of this approach is the use of pulsed coupled oscillators that concurrently perform synchronisation and desynchronisation across multiple channels. This allows for rapid convergence to the steady state in a completely decentralised manner, that is, without requiring a node or channel coordinator, or time synchronisation via a global time synchronisation mechanism. DT-SCS spontaneously adapts to node churn and varying packet losses, while offering high degree of connectivity through channel swapping. Experimentation via simulations and a real Contiki-OS based implementation on TelosB motes shows that, in comparison to TSCH and the EM-MAC protocol, the proposed DT-SCS protocol leads to a significant reduction of the convergence time and substantially higher network throughput use. These traits render the proposed DT-SCS protocol an excellent candidate for vehicular or mobile deployments that collect and communicate large quantities of information

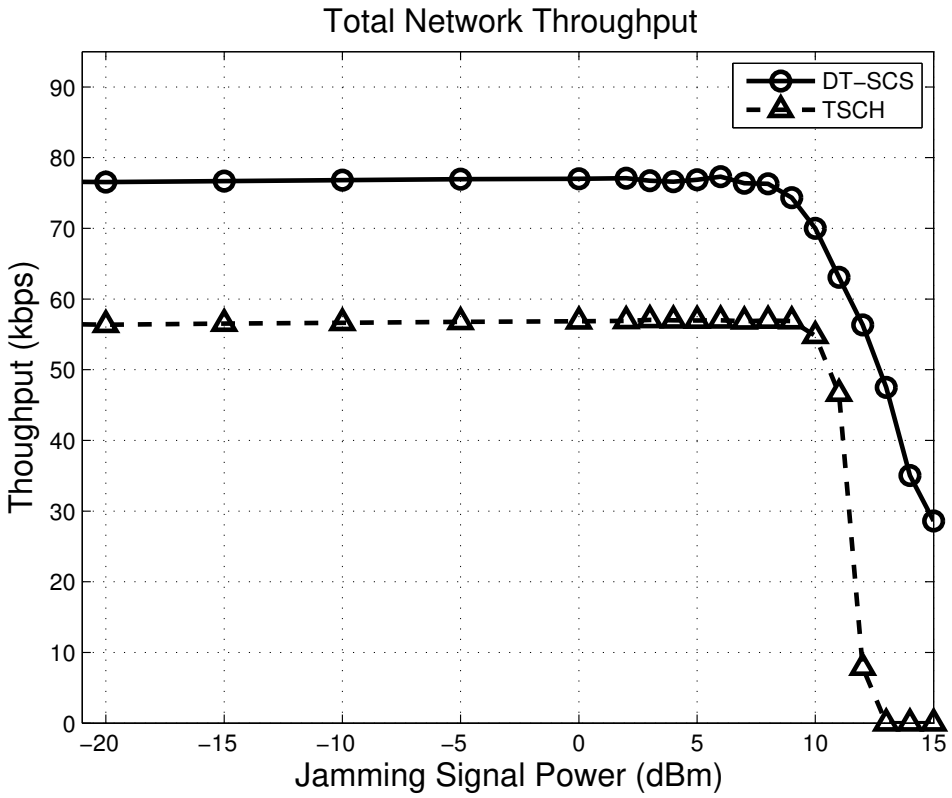


Figure 5.14: Total network bandwidth (total payload bits transmitted by all nodes per second) between the proposed DT-SCS and TSCH under varying signal power levels.

in a decentralised manner.

In certain circumstances, it is believed that TSCH would outperform the proposed DT-SCS. One such circumstance example exists for networks with more complex topologies where the assumption of a densely-meshed network does not hold. Here, DT-SCS may struggle with the initial stages of balancing, as nodes can only react to packets they hear. Multiple nodes may transmit at the same time, unaware of each others existence in the channel, resulting in packet collisions. If a solution exists where nodes can position themselves in a channel where all neighbours have good connectivity, then the protocol will find this; however, in sparsely connected networks this solution may take a disproportionate time to find, or may not exist at all and DT-SCS may never reach a steady state. Moreover, TSCH would continue to function in such a scenario, since only nodes with connectivity can exchange RQ/ACK packets to establish connectivity.

As presented in this chapter, DT-SCS only allows for a fair and balanced TDMA timeslot structure to be created, that is, with all nodes being given equal bandwidth share within a channel. In a real deployment, it may be preferred that some nodes have a larger share of the available bandwidth. Using the PCO approach would make it difficult for nodes to surrender a fraction of their timeslot for use by other nodes. When there are large numbers of nodes per

channel, the amount of time available for sending data per node becomes much reduced, due to the fixed guard times required for beacon variability and channel swapping.

The DT-SCS Contiki-OS code can be downloaded from the experiment online repository, https://github.com/m1geo/DTSCS, and is also included for completeness in Appendix 7.3.2.

Chapter 6

Conclusions and Future Work

This work looks at energy awareness in multichannel MAC protocols with specific regard to Wireless Sensor Networks amenable to Internet-of-Things oriented deployments. The work has a practical focus with hardware experiments using Crossbow TelosB nodes running Contiki-OS. The work has three main contributions:

- Experimental study and analysis of energy harvesting possibilities
- A novel energy management framework for energy-neutral operations in multichannel multi-tier WSNs suitable for IoT-oriented communications
- A new distributed multichannel protocol that limits idle listening and ensures quick convergence without the need for a centralised time synchronisation

In Chapter 3, experiments and analysis of energy production rates for common energy harvesting technologies were considered to provide an initial coverage of the link between the recent flurry of literature on probability models describing the energy harvesting process and the very limited experimental evidence supporting such models. Motivated by the lack of such experimental evidence on the capabilities of practical transducer deployments, the approach here was to report on empirical observations with data collected using a purpose built multi-transducer energy harvesting testbed, deployed in varied locations. The software tools and hardware design notes are provided, along with a full dataset that can be used for future research, all of which can be found online at `http://github.com/m1geo/EH_IOT`.

In Chapter 4 the concept of energy neutrality was explored in detail, in order to develop analytic conditions for perpetual energy autonomy per sensor in a typical TDMA network. The chapter proposed an analytic framework for characterising practical energy neutrality uniformly-formed WSNs. The importance of the application data transmission rate in the network's energy dissipation was shown, providing an analytic assessment of the expected energy dissipation as

a function of the system parameters, under a variety of statistical characterisations for the data transmission rate of each sensor node. The analytic framework is validated using experimental assessment with a typical collision-free low-energy rapid convergence MAC protocol implemented on TelosB nodes running Contiki-OS. An accuracy of within 7% is observed between the framework and measured energy consumption. This analysis could easily be used for a particular harvesting technology, such as those presented in Chapter 3 to predict the smallest possible energy harvesting interval required for energy-neutral deployment, or, to find the best possible data transmission rate that can be accommodated for a given set of system parameters. This would save on costly and cumbersome testing in the field.

Chapter 5 proposed a novel MAC protocol for IoT and WSN nodes that performs decentralised time-synchronised channel swapping (DT-SCS) and circumvents certain convergence and network utilisation problems of existing MAC designs, such as the state-of-the-art TSCH protocol, now part of the IEEE 802.15.4e-2012 standard [81]. Specifically, the unique aspect of the approach was the use of pulsed coupled oscillators that concurrently perform synchronisation and desynchronisation across multiple channels, allowing for rapid convergence to steady-state in a completely decentralised manner, that is, without requiring a coordinating node or channel, or time synchronisation via a global clock. DT-SCS adapts automatically to noise in the network caused by packet loss or node churn, while features such as channel swapping allow for a high degree of connectivity. Experimentation via simulation and real Contiki-OS based implementation on TelosB motes show that DT-SCS has a significantly shorter convergence time and substantially higher network bandwidth when compared to TSCH and EM-MAC. These traits render DT-SCS an excellent candidate for many kinds of mobile and vehicular deployments where large amounts of information are moved in a decentralised manner.

6.1 Future Work

Considering extensions of the empirical study in Chapter 3, it may be interesting to deploy the sensor in more locations and for longer periods of time. In doing so, it may be possible to model longer time scale trends, such as how harvested energy changes between seasons. It may be possible to better fit the harvested energy data by considering more distributions when generating the mixture models or carrying out multiple fitting experiments for different time intervals. Several broad unexplored sources of energy exist, such as radio-frequency (RF) electromagnetic radiation and thermal gradients, which are scarcely mentioned in the literature. Since the technology for harvesting electrical energy from these sources is readily available, e.g., the Seebeck effect for a thermal gradient, characterisation and modelling of other sources would

add directly to the available literature. Another interesting alternative would be to replace the simple load resistance with a dynamic load in order to better understand the available energy and improve the system sensitivity in poorer harvesting conditions. Extending the platform to include energy storage, such as rechargeable batteries and, potentially, double energy stores (where the rechargeable battery is coupled with a super-capacitor) may yield interesting results. Regarding the development of the harvesting platform itself, addition of a real-time clock IC would allow for absolute timestamps as opposed to relative timestamps provided by the MCU's internal `millis()` function, which is known to be inaccurate over long periods due to manufacturing tolerances in the MCU's clock crystal frequency.

The work in Chapter 4 may be extended to consider the physical locations of sensor nodes, considering the distances involved, and the required transmit power of the sending node to achieve an acceptable SNR. Work by Ayinde *et al.* [148] already considers physical locations of nodes in order to minimise relay nodes, which inevitably have a higher energy demand. Redondi *et al.* [149] used a similar model to derive results for energy-optimal spatio-temporal coverage parameters of VSNs, exploring the tradeoff with the incurred energy consumption for well-known video bitstreams. Renna *et al.* [150] also used a similar model to minimise the energy consumption of IoT devices performing feature extraction and to reduce the cost of cloud infrastructure billing at the computing service receiving the extracted features.

Further study of how the DT-SCS protocol in Chapter 5 performs under sparsely connected conditions could make for an interesting future work topic. By viewing the nodes within each DT-SCS channel fragment (that is, nodes on the same channel but not within communication range) as beads on a ring, as in Figure 5.2, with each group of fragmented nodes a separate ring, it may be possible to form an analogy to between the beads (nodes) on each ring (channel fragment) as teeth on a mechanical gear. Nodes appearing in multiple channel fragments could be considered as anchored SYNC nodes, similar to the anchored desynchronisation work by Lien *et al.* [145]. In such a case, a single anchored node would simply replace the standard SYNC node described in Chapter 5, performing all of the required functions. However, multiple anchored nodes would cause bunching of the normal (free-moving) nodes between the anchored nodes. An idea for resolving this could be to have a rule similar to the channel swapping algorithm of Subsection 5.2.3.4; if moving the other side of this anchored SYNC node would allow for a larger TDMA slot, then move, with an appropriate check to prevent race conditions.

It would be interesting to address the case where the deployment application requires network bandwidth to be unevenly distributed between nodes in channels, as mentioned in Section 5.6. One simple solution could be to amend the protocol such that each node beacons multiple times at equally spaced intervals throughout the period T. Then, using a slightly modified

beacon packet structure, a node could offer unused timeslots to other nodes.

The protocol may further be improved to specify an algorithm for adjusting the protocol guard times and listening intervals dynamically, in order to further reduce energy. In the current form, DESYNC nodes power their radios at a fixed time interval before the expected beacon, corresponding to the listening interval and wait for a specified guard time following the transmission of their beacon packet. It may prove more beneficial to have a variable guard and listening times which gradually adjust towards a minimum energy state. Equivalently, SYNC nodes could gradually reduce their listening interval to reduce their energy demands, given that once steady state operation has been achieved, any PCO updates are likely to be small.

Since nodes in a multihop network can appear as hidden nodes, the existing hidden node simulations may provide an understanding of how the network could operate. Problems arise when nodes oscillate in and out of range, as this presents itself as churn, effecting network bandwidth and stability. Two potential solutions exist: *(i)* blacklisting nodes from settling in channels where nodes are known to be on the threshold, or *(ii)* persistent swapping such that nodes are constantly moving, and nodes out of range are akin to packet loss and absorbed by the coupling parameters. Once DT-SCS can handle multihop typologies, interfacing with higher layers (see Figure 2.1) will provide for packet routing and forwarding in multihop networks.

Chapter 7

Appendix

The appendix is split into two sections: Section 7.1 relates to Chapter 4, while Section 7.2 relates to Chapter 5.

7.1 Appendix I: Proofs of Maximum Residual Energy

This section of the Appendix relates to Chapter 4: Analytic Conditions for Energy Neutrality in Uniformly-formed WSNs.

For all distributions, the first derivative is provided and it is shown that, when set to zero and under $n \in (0, \infty)$, this leads to a single admissible extremum value. Under this value, it is demonstrated that the second derivative is guaranteed to be negative. Thus, the extremum value maximises the residual energy under each data transmission rate PDF.

7.1.1 Pareto distribution

The first derivative of $E_{n,\mathrm{P}}$ to n, $n \in (0, \infty)$, is

$$\frac{dE_{n,\mathrm{P}}}{dn} = cT \left[\frac{ab}{n^2} - \frac{a}{n^2} (b+p) \left(\frac{vn}{a} \right)^{\alpha} \right]. \tag{7.1}$$

Assuming that $b \neq 0$, the only admissible solution of $\frac{dE_{n,\mathrm{P}}}{dn} = 0$ is given in (4.15), as all other solutions are complex numbers. In conjection with the fact that $E_{n,\mathrm{P}}$ is differentiable for $n \in (0, \infty)$, this demonstrates that $n_{0,\mathrm{P}}$ is the global extremum or inflection point of $E_{n,\mathrm{P}}$. The second derivative of $E_{n,\mathrm{P}}$ is

$$\frac{d^2 E_{n,\mathrm{P}}}{dn^2} = cT \left[-\frac{2ab}{n^3} - \frac{a}{n^3} (\alpha - 2)(b+p) \left(\frac{vn}{a} \right)^{\alpha} \right]. \tag{7.2}$$

By evaluating $\frac{d^2E_{n,\mathrm{P}}}{dn^2}$ for $n_{0,\mathrm{P}}$ nodes, one obtains

$$\frac{d^2E_{n,\mathrm{P}}}{dn^2}\left(n_{0,\mathrm{P}}\right) = -\frac{cT\left(\left[2b+(\alpha-2)\right](b+p)^{\frac{3}{\alpha}}v^3\right)}{a^2b^{\frac{3}{\alpha}}}, \tag{7.3}$$

which is negative since $\alpha \geq 2$ and all variables are positive. This means that the maximum possible residual energy for $n \in (0,\infty)$ is achieved under $n = n_{0,\mathrm{P}}$. This derivation also covers the case of fixed-rate data production and transmission if $v = (d+1)(r-1)$ and $\alpha = r$ are set.

7.1.2 Exponential distribution

The first derivative of $E_{n,\mathrm{E}}$ to n, $n \in (0,\infty)$, is

$$\frac{dE_{n,\mathrm{E}}}{dn} = cT\left[\frac{ab}{n^2} - \frac{a}{n^2}(b+p)\exp\left(-\frac{a}{n(d+1)r}\right)\right]. \tag{7.4}$$

Assuming that $b \neq 0$, the only admissible solution for $\frac{dE_{n,\mathrm{E}}}{dn} = 0$ is given in (4.21), as all other solutions are complex numbers. In conjunction with the fact that $E_{n,\mathrm{E}}$ is differentiable for $n \in (0,\infty)$, this demonstrates that $n_{0,\mathrm{E}}$ is the global extremum or inflection point of $E_{n,\mathrm{E}}$. The second derivative of $E_{n,\mathrm{E}}$ is

$$\begin{aligned}\frac{d^2E_{n,\mathrm{E}}}{dn^2} &= cT\left[-\frac{2ab}{n^3} - \frac{a}{n^4(d+1)r}(b+p)\right.\\ &\times \left.\exp\left(-\frac{a}{n(d+1)r}\right)\left[a-2n(d+1)r\right]\right]\end{aligned} \tag{7.5}$$

By evaluating $\frac{d^2E_{n,\mathrm{E}}}{dn^2}$ for $n_{0,\mathrm{E}}$ nodes, one obtains

$$\frac{d^2E_{n,\mathrm{E}}}{dn^2}\left(n_{0,\mathrm{E}}\right) = -\frac{cT(d+1)^3br^3}{a^2}\left[\ln\left(\frac{b+p}{b}\right)\right]^4, \tag{7.6}$$

which is negative since all variables are positive and the natural logarithm is raised to an even power. This means that the maximum possible residual energy for $n \in (0,\infty)$ is achieved under $n = n_{0,\mathrm{E}}$.

7.1.3 Half-Gaussian distribution

The first derivative of $E_{n,\mathrm{H}}$ to n, $n \in (0,\infty)$, is

$$\frac{dE_{n,\mathrm{H}}}{dn} = cT\left[-\frac{ap}{n^2} + \frac{a}{n^2}(b+p)\operatorname{erf}\left(\frac{a}{\sqrt{\pi}(d+1)rn}\right)\right]. \tag{7.7}$$

Assuming that $p \neq 0$, the only admissible solution for $\frac{dE_{n,\mathrm{H}}}{dn} = 0$ is given in (4.25). In conjunction with the fact that $E_{n,\mathrm{H}}$ is differentiable for $n \in (0,\infty)$, this demonstrates that $n_{0,\mathrm{H}}$ is the

global extremum or inflection point of $E_{n,\mathrm{H}}$. Then, the second derivative of $E_{n,\mathrm{H}}$ is

$$\begin{aligned}\frac{d^2E_{n,\mathrm{H}}}{dn^2} &= cT\left[\frac{2ap}{n^3}-\frac{2a}{n^3}\left(b+p\right)\right.\\ &\times \operatorname{erf}\left(\frac{a}{\sqrt{\pi}(d+1)rn}\right)-\frac{2a^2}{\pi(d+1)rn^4}\\ &\times \left.\left(b+p\right)\exp\left(-\frac{a^2}{\pi((d+1)r)^2n^2}\right)\right].\end{aligned} \tag{7.8}$$

By evaluating $\frac{d^2E_{n,\mathrm{H}}}{dn^2}$ for $n_{0,\mathrm{H}}$ nodes, one obtains

$$\begin{aligned}\frac{d^2E_{n,\mathrm{H}}}{dn^2}\left(n_{0,\mathrm{H}}\right) &= -\frac{2\pi cT\left(d+1\right)^3r^3\left(b+p\right)}{a^2}\\ &\times \left[\operatorname{erf}^{-1}\left(\frac{p}{b+p}\right)\right]^4\\ &\times \exp\left(-\left[\operatorname{erf}^{-1}\left(\frac{p}{b+p}\right)\right]^2\right),\end{aligned} \tag{7.9}$$

which is negative since the inverse error function is raised to an even power and all variables are positive. This means that the maximum possible residual energy for $n\in(0,\infty)$ is achieved under $n=n_{0,\mathrm{H}}$.

7.1.4 Gaussian distribution

The first derivative of $E_{n,\mathrm{N}}$ to n, $n\in(0,\infty)$, is

$$\begin{aligned}\frac{dE_{n,\mathrm{N}}}{dn} &= cT\left[-\frac{ap}{n^2}+\frac{a}{2n^2}\left(b+p\right)\left[\operatorname{erf}\left(\frac{\left(d+1\right)r}{\sqrt{2}\sigma}\right)\right.\right.\\ &- \left.\left.\operatorname{erf}\left(\frac{\left(d+1\right)r-\frac{a}{n}}{\sqrt{2}\sigma}\right)\right]\right].\end{aligned} \tag{7.10}$$

Assuming that $p\neq 0$, the only admissible solution for $\frac{dE_{n,\mathrm{N}}}{dn}=0$ is given in (4.30) In conjunction with the fact that $E_{n,\mathrm{N}}$ is differentiable for $n\in(0,\infty)$, $n_{0,\mathrm{N}}$ corresponds to the global extremum or inflection point of $E_{n,\mathrm{N}}$. Then, the second derivative of $E_{n,\mathrm{N}}$ is

$$\begin{aligned}\frac{d^2E_{n,\mathrm{N}}}{dn^2} &= cT\left[\frac{2ap}{n^3}+\frac{a}{n^3}\left(b+p\right)\right.\\ &\times \left[\operatorname{erf}\left(\frac{\left(d+1\right)r-\frac{a}{n}}{\sqrt{2}\sigma}\right)-\operatorname{erf}\left(\frac{\left(d+1\right)r}{\sqrt{2}\sigma}\right)\right]\\ &- \left.\frac{a^2}{\sqrt{2\pi}\sigma n^4}\left(b+p\right)\exp\left(\frac{-\left[\left(d+1\right)r-\frac{a}{n}\right]^2}{2\sigma^2}\right)\right].\end{aligned} \tag{7.11}$$

By evaluating $\frac{d^2 E_{n,\mathrm{N}}}{dn^2}$ for $n_{0,\mathrm{N}}$ nodes, one obtains

$$\frac{d^2 E_{n,\mathrm{N}}}{dn^2}\left(n_{0,\mathrm{N}}\right) = -\frac{cT\left(b+p\right)}{a^2\sqrt{2\pi}\sigma}\left[\left(d+1\right)r - \sqrt{2}\sigma c_{\mathrm{N}}\right]^4 \exp\left(-c_{\mathrm{N}}^2\right), \tag{7.12}$$

which is negative since $\left[\left(d+1\right)r - \sqrt{2}\sigma c_{\mathrm{N}}\right]$ is raised to an even power and all variables are positive. This means that the maximum possible residual energy for $n \in (0,\infty)$ is achieved under $n = n_{0,\mathrm{N}}$.

7.2 Appendix II: Proofs of DT-SCS Propositions

This section of the Appendix relates to Chapter 5: Decentralised Time-Synchronised Channel Swapping for Ad Hoc Networks.

7.2.1 Proof of Proposition 5.1: Balancing

During the `Converging` mode, a SYNC node may switch from channel c to $c+1$, or from channel $c-1$ to c. A SYNC node switch occurring simultaneously between channels $c-1 \to c$ and $c \to c+1$ at the kth period can be expressed stochastically for the number of nodes in channel $c \in \{1,\ldots,C\}$ by

$$\overline{W}_c^{(k+1)} = \overline{W}_c^{(k)} - u\left[\overline{W}_c^{(k)} - \overline{W}_{c+1}^{(k)} - 1\right] p_{c+1}^{(k)} \overline{W}_c^{(k)} + u\left[\overline{W}_{c-1}^{(k)} - \overline{W}_c^{(k)} - 1\right] p_c^{(k)} \overline{W}_{c-1}^{(k)}, \tag{7.13}$$

while for channel C,

$$\overline{W}_C^{(k+1)} = \overline{W}_C^{(k)} - u\left[\overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2\right] p_1^{(k)} \overline{W}_C^{(k)} + u\left[\overline{W}_{C-1}^{(k)} - \overline{W}_C^{(k)} - 1\right] p_C^{(k)} \overline{W}_{C-1}^{(k)} \tag{7.14}$$

and for channel 1,

$$\overline{W}_1^{(k+1)} = \overline{W}_1^{(k)} - u\left[\overline{W}_1^{(k)} - \overline{W}_2^{(k)} - 1\right] p_2^{(k)} \overline{W}_1^{(k)} + u\left[\overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2\right] p_1^{(k)} \overline{W}_C^{(k)}, \tag{7.15}$$

where $u[\cdot]$ is the unit-step function, $\overline{W}_{c-1}^{(k)}$, $\overline{W}_c^{(k)}$ and $\overline{W}_{c+1}^{(k)}$ are the expected numbers of nodes at channels $c-1$, c and $c+1$ during the kth period, and $p_c^{(k)}$ is the probability the SYNC node will successfully switch to channel c during the k period. A remark that $p_c^{(k)}$ is smaller

than unity since (typically) only a single (SYNC) node will switch channels or, in the case of interference, no node will manage to switch.

For every channel $c \in \{1, \dots, C\}$ the transition system formed by (7.13) is written in matrix form as

$$\overline{\mathbf{w}}^{(k+1)} = \mathbf{G}^{(k)} \overline{\mathbf{w}}^{(k)} \tag{7.16}$$

with

$$\overline{\mathbf{w}}^{(k+1)} = \left[\overline{W}_1^{(k+1)} \; \overline{W}_2^{(k+1)} \; \cdots \; \overline{W}_{C-1}^{(k+1)} \; \overline{W}_C^{(k+1)} \right]^T, \tag{7.17}$$

$$\overline{\mathbf{w}}^{(k)} = \left[\overline{W}_1^{(k)} \; \overline{W}_2^{(k)} \; \cdots \; \overline{W}_{C-1}^{(k)} \; \overline{W}_C^{(k)} \right]^T \tag{7.18}$$

and

$$\mathbf{G}^{(k)} = \begin{bmatrix} 1 - g_1^{(k)} & 0 & 0 & \cdots & g_C^{(k)} \\ g_1^{(k)} & 1 - g_2^{(k)} & 0 & \cdots & 0 \\ 0 & g_2^{(k)} & \ddots & \cdots & 0 \\ \vdots & \vdots & \ddots & 1 - g_{C-1}^{(k)} & 0 \\ 0 & 0 & 0 & g_{C-1}^{(k)} & 1 - g_C^{(k)} \end{bmatrix} \tag{7.19}$$

where $\forall c < C: \; g_c^{(k)} = u\left[\overline{W}_c^{(k)} - \overline{W}_{c+1}^{(k)} - 1\right] p_{c+1}^{(k)}$ and $g_C^{(k)} = u\left[\overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2\right] p_1^{(k)}$.

The eigenvectors of the system in (7.16) are given by

$$\mathbf{w}^{(\mathrm{SS})} = \begin{bmatrix} \lfloor \frac{W}{C} \rfloor & \cdots & \lceil \frac{W}{C} \rceil \end{bmatrix}^T. \tag{7.20}$$

This is because $\mathbf{w}^{(\mathrm{SS})}$ vectors in the form of (7.20) lead to

$$\begin{aligned} \forall c: & \begin{cases} u\left[\overline{W}_c^{(\mathrm{SS})} - \overline{W}_{c+1}^{(\mathrm{SS})} - 1\right] = 0 \\ u\left[\overline{W}_C^{(\mathrm{SS})} - \overline{W}_1^{(\mathrm{SS})} - 2\right] = 0 \end{cases} \\ \Rightarrow \forall c: & \; g_c^{(\mathrm{SS})} = 0. \end{aligned} \tag{7.21}$$

Thus,

$$\forall c: \; \lim_{k \to \infty} W_c^{(k)} \in \left\{ \left\lceil \frac{W}{C} \right\rceil, \left\lfloor \frac{W}{C} \right\rfloor \right\}. \tag{7.22}$$

Note that the transition matrix $\mathbf{G}^{(k)}$ in (7.19) has all its columns summing to unity, while its entries are non-negative and smaller than unity. As such, via the Perron–Frobenius theorem [151], the maximum magnitude of all eigenvalues of $\mathbf{G}^{(k)}$ is unity, that is, all eigenvalues of any instantiation of $\mathbf{G}$ are within (or on) the unit circle. Therefore, the system in (7.16) will reach a vector of the form of (7.20), or will oscillate between multiple of these vectors (fixed points). However, no such oscillations can occur, since, due to the switching rules of (5.4) and (5.5),

all higher-numbered channels must contain $\lceil \frac{W}{C} \rceil$ nodes and all lower-numbered channels must contain $\lfloor \frac{W}{C} \rfloor$ nodes. Thus, the proposed balancing mechanism converges to a single fixed point in (7.20).

7.2.2 Proof of Proposition 5.2: Stability

PCO-based synchronisation is well-known to achieve convergence [33]. Hence, during the `Converging` mode, all SYNC nodes will converge to synchronous beacons across all C channels, given that their beacon packet broadcasts are only affected by other SYNC node broadcasts. PCO-based desynchronisation within each channel is then equivalent to anchored desynchronisation [145]. The latter is proven to converge to a steady state wherein the packet broadcasts are equidistant within the beacon period, i.e., at intervals of $T\left(\frac{1}{W_c} \pm b_{\text{thres}}\right)$ seconds. Once this is achieved and all nodes are balanced across all channels (the latter is ensured via Proposition 5.1), the system moves to `Converged` mode.

Channel swapping events do not affect the converged beacon packet transmissions within each channel since: *(i)* nodes between unbalanced channels cannot perform swaps; *(ii)* swapping requests and acknowledgements are done in the guard time periods; *(iii)* once swapping is acknowledged, nodes broadcast their first beacon packet in their new channel at the end of the guard period. In this way, they can confirm that the node they are swapping with has left the channel. Selecting the post-beacon guard period to be smaller than $b_{\text{thres}}T$ seconds ensures that the convergence is not disturbed by channel swapping.

7.2.3 Proof of Proposition 5.3: Connectivity

The average degree of connectivity of a DESYNC node is given by the total number of connections established by DESYNC nodes divided by the total number of DESYNC nodes (i.e., $W - C$). The total number of connections is found by multiplying the number of DESYNC nodes with the number of connections established by each of them. Particularly, each DESYNC node in a channel can connect to *(i)* all the SYNC nodes, *(ii)* the remaining DESYNC nodes in the same channel and *(iii)* the DESYNC nodes in other balanced channels (i.e., channels with the same number of nodes) that do not have the same phase. Hence, the number of connections established by DESYNC nodes in the highest and lowest channels is

$$C_{\text{high}} W_{\text{DESYNC,high}} \times (C_{\text{high}} W_{\text{DESYNC,high}} + C_{\text{low}})$$

and

$$C_{\text{low}} W_{\text{DESYNC,low}} \times (C_{\text{low}} W_{\text{DESYNC,low}} + C_{\text{high}}) \,,$$

respectively. Summing the above expressions and dividing by $W - C$ leads to (5.11).

7.2.4 Proof of Proposition 5.4: Convergence Time

To derive the possible combinations of W nodes in C channels, one must begin by assuming zero nodes in channels $1, 2, \ldots, C-1$; this means that all W nodes must be in channel C. If zero nodes exist in channels $1, 2, \ldots, C-2$ and one node exists in channel $C-1$, this means that $W-1$ nodes must be in channel C. Continuing this expansion, all possible cases (two nodes in channel $C-1$ and $W-2$ nodes in channel C and so on) are covered. For the non-trivial case of $C \geq 2$ and $W \geq 2C$, this leads to the following summation:

$$\mathcal{C}_{W,C} = \sum_{i_1=0}^{W} \sum_{i_2=0}^{W-i_1} \cdots \sum_{i_{C-2}=0}^{W-\sum_{j=1}^{C-3}} \left(W - \sum_{j=1}^{C-2} i_j + 1 \right). \tag{7.23}$$

By calculating the result of the series of (7.23), one reaches (5.13).

Since nodes join a channel randomly, once each node makes a decision, it is a "success" or "fail" process for each channel: "success" if the node joins it, "fail" otherwise. The probability of "success" is $\frac{1}{C}$, while the probability of "fail" is $\frac{C-1}{C}$. Hence, for the first channel, the probability of combination i (out of $\mathcal{C}_{W,C}$) having $W_1(i)$ nodes ("successes") out of W (based on the binomial distribution) is:

$$\Pr(i, \text{Ch1}) = \binom{W}{W_1(i)} \frac{(C-1)^{W-W_1(i)}}{C^W}. \tag{7.24}$$

For the second channel, the probability of combination i having $W_2(i)$ nodes out of $W - W_1(i)$ possible nodes [assuming that $W_1(i)$ nodes have chosen to join the first channel] is:

$$\Pr(i, \text{Ch2}) = \binom{W - W_1(i)}{W_2(i)} \frac{(C-1)^{W-W_1(i)-W_2(i)}}{C^{W-W_1(i)}}. \tag{7.25}$$

Iterating this for all channels, one can derive in a similar fashion $\Pr(i, \text{Ch3}), \ldots \Pr(i, \text{Ch}\{C-1\})$. The remaining number of nodes, i.e., $\left[W - \sum_{c=1}^{C-1} W_c(i)\right]$ nodes, will be joining channel C with probability $\Pr(i, \text{Ch}\{C\}) = 1$. Since these probabilities are independent, the probability of combination i having the node distribution: $[W_1(i) \ldots W_C(i)]$ in channels $1, \ldots, C$ is given by (5.14). Notice that the assumption of nodes deciding first on whether to join channel 1, then whether to join channel 2, and so on, is not restrictive. In fact, the above analysis can be expressed with any order of channels without affecting the result. In other words, the numbering of channels stated above has no effect on $\Pr(i)$.

It is then possible to estimate the expected delay until convergence via (5.12), with the

expression in the maximisation of (5.12) establishing the largest imbalance of the node distribution of combination i from the average number of nodes per channel, $\lfloor \frac{W}{C} \rceil$. This expresses the channel that will experience the highest number of channel switches until convergence (each requiring N_e periods for SYNC node election and N_c).

7.3 Appendix III: Code Listings

Code written by the author during the course of this PhD has been included in this section of the Appendix for completeness. However, the author suggests that you download the code from the GitHub repositories linked throughout the text and again below to ensure you have the latest version of the code, to avoid typing errors, and to save your fingers!

7.3.1 Energy Harvester Platform

All of the code in this section can be downloaded from `http://github.com/m1geo/EH_IOT`.

7.3.1.1 Arduino Code

This code was written for the Arduino IDE 1.6.11 for the Arduino Uno platform with a custom designed shield (see Figure 3.2 and Figure 3.3), with SD card, a DHT11 temperature (and humidity) sensor and BH1750 ambient light sensor.

The main C file run on the Arduino is `EH_IOT.ino`. The code contains some standard initialisation code to set the pin status, initialise the I^2C and SPI buses and configure the SD Card. The code creates a new CSV file on each run, so as not to clobber existing data. The time-stamp is referenced from the Arduino's own clock [i.e., `millis()`] as the RTC hardware is not used.

```
/*
 Energy Harvesting For The Internet-of-Things:
     Measurements And Probability Models

 George Smart <g.smart@ee.ucl.ac.uk>
 John Atkinson <john.atkinson.10@ucl.ac.uk>

 Electronic & Electrical Engineering, University College London, UK.

 13 July 2015

SD Card Interconnects attach to SPI the bus as follows:
 - MOSI    - pin 11
```

```
  - MISO     - pin 12
  - CLK      - pin 13
  - CS       - pin 10

 Analog Sensors Interconnects:
  - Solar    - pin A0
  - Diode    - pin A1
  - Thermal  - pin A2
  - Piezo    - pin A3

 Envronment Sensors:
  - DHT11    - pin 2
  - BH1750   - I2C Bus (A4/A5)

 Debug LEDs:
  - Solar    - pin 4
  - Diode    - pin 5
  - Thermal  - pin 6
  - Piezo    - pin 7
*/

#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <BH1750.h>
#include "DHT.h"

// Ethernet Shield, CS = pin 4.
// Standard SD Sheild, CS = pin 10.
const int chipSelect = 10;

#define DHTPIN 2
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
BH1750 lightMeter;

uint8_t sensor_read = 0;
unsigned long currMeasure = 0;
char filename[] = "LOGGER00.CSV"; // auto set later to be 00-99.

#define DHT    2
#define SDACT  3
#define LEDA   4
#define LEDB   5
```

```
#define LEDC    6
#define LEDD    7

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  lightMeter.begin();
  dht.begin();

  Serial.print("Initialising SD card...");
  pinMode(chipSelect, OUTPUT);
  pinMode(SDACT, OUTPUT);
  pinMode(DHT,    INPUT);
  pinMode(LEDA,   OUTPUT);
  pinMode(LEDB,   OUTPUT);
  pinMode(LEDC,   OUTPUT);
  pinMode(LEDD,   OUTPUT);

  // see if the card is present and can be initialised:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("Card initialised.");

  // find the lowest filename that's not used.
  for (uint8_t i = 0; i < 100; i++) {
   filename[6] = i/10 + '0';
   filename[7] = i%10 + '0';
   if (!SD.exists(filename)) {
    break;
   }
  }

  Serial.print("Logging to: ");
  Serial.println(filename);

  digitalWrite(LEDA, HIGH);
```

```
  digitalWrite(LEDB, HIGH);
  digitalWrite(LEDC, HIGH);
  digitalWrite(LEDD, HIGH);
  delay(250);

  // write CSV column headers
  File dataFile = SD.open(filename, FILE_WRITE);
  dataFile.println("msTime,Solar,Diode,Therm,Piezo,LightMeter,TempMeter,
    HumiMeter");
  dataFile.close();

  // Strobe output LEDs
  digitalWrite(LEDA, LOW);
    delay(100);
    digitalWrite(LEDA, HIGH);
  digitalWrite(LEDB, LOW);
    delay(100);
    digitalWrite(LEDB, HIGH);
  digitalWrite(LEDC, LOW);
    delay(100);
    digitalWrite(LEDC, HIGH);
  digitalWrite(LEDD, LOW);
    delay(100);
    digitalWrite(LEDD, HIGH);
}

uint16_t lux = 0;
double humi = 0;
double temp = 0;

void loop()
{
  currMeasure = millis();
  // force sensor reading every 300 readings (30 seconds)
  if (sensor_read == 0) {
    lux = lightMeter.readLightLevel();
    humi = dht.readHumidity();  //DHT11 is very slow
    temp = dht.readTemperature();
  }
  sensor_read++;
  if (sensor_read >= 20) {
    sensor_read = 0;
  }
```

```
145   int Solar = analogRead(A0);
      int Diode = analogRead(A1);
147   int Therm = analogRead(A2);
      int Piezo = analogRead(A3);
149
      if (Solar > 0) {digitalWrite(LEDA, LOW);} else {digitalWrite(LEDA, HIGH);}
151   if (Diode > 0) {digitalWrite(LEDB, LOW);} else {digitalWrite(LEDB, HIGH);}
      if (Therm > 0) {digitalWrite(LEDC, LOW);} else {digitalWrite(LEDC, HIGH);}
153   if (Piezo > 0) {digitalWrite(LEDD, LOW);} else {digitalWrite(LEDD, HIGH);}

155   // make a string for assembling the data to log:
      String dataString = "";
157   dataString += String(currMeasure);
      dataString += ",";
159   dataString += String(Solar);
      dataString += ",";
161   dataString += String(Diode);
      dataString += ",";
163   dataString += String(Therm);
      dataString += ",";
165   dataString += String(Piezo);
      dataString += ",";
167   dataString += String(lux);
      dataString += ",";
169   dataString += String((int)temp);
      dataString += ",";
171   dataString += String((int)humi);

173   // open the file. note that only one file can be open at a time,
      // so you have to close this one before opening another.
175   File dataFile = SD.open(filename, FILE_WRITE);

177   // if the file is available, write to it:
      if (dataFile) {
179     digitalWrite(SDACT, HIGH);
        dataFile.println(dataString);
181     dataFile.close();
        digitalWrite(SDACT, LOW);
183
        Serial.println(dataString);
185   }
      else {
187     Serial.print("error opening ");
        Serial.println(filename);
```

```
189   }

191   // wait until the next measurement time...
      while (millis() < (currMeasure + 100));
193 }
```

Code/EH_IOT.ino

7.3.2 DT-SCS Contiki-OS Code

All of the code in this section can be downloaded from https://github.com/m1geo/DTSCS, under the "`Contiki Code`" section. Please see Subsection 7.3.2.2 for details on patching the `rtimer` library in Contiki-OS 2.x in order to allow multiple updates before timer firing.

7.3.2.1 Contiki-OS TelosB DT-SCS Application Code

The following listing is a lightweight implementation of the Decentralised Time-Synchronised Channel Swapping (DT-SCS) MAC protocol for Ad Hoc Wireless Networks. The code is written in C for the Contiki-OS open source operating system, version 2.7, with TelosB (T-mote Sky) wireless sensor motes. It is built as an application layer program inside Contiki-OS for ease of implementation and future understanding. To this end, the standard Contiki-MAC and radio duty cycler are disabled and, instead, handle these functions inside the DT-SCS application code. To allow this pass-through behaviour, we use the *nullmac* and *nullrdc* drivers, which are selected inside `project-conf.h`. The the `rtimer` library must be patched (see Subsection 7.3.2.2).

```
/* DT-SCS Light : Desync with Election & Sync.
2  * DTSCS_light.c (PhD Thesis Copy)
   *
4  * Project:    https://github.com/m1geo/DTSCS
   *
6  * Author:     http://www.george-smart.co.uk/
   *         http://wwws.ee.ucl.ac.uk/~zceed42/
8  *
   * George Smart <g.smart@ee.ucl.ac.uk> (PhD Student)
10 * Yiannis Andreopoulos <iandreop@ee.ucl.ac.uk> (Supervisor)
   *
12 * Monday 08 January 2016.  2011-2017 -- M1GEO.
   *
14 * Telecommunications Research Group Office, Room 7.06, Desk 162.
   * Malet Place Engineering Buidling
```

```
 * Department of Electronic & Electrical Engineering
 * University College London
 * Malet Place, London, WC1E 7JE, United Kingdom.
 *
 * Issues:
 *  - election favours the higher node, instead of random roll
 *  - nodes are hard coded per channel (no balancing in light version)
 *  - code is a little buggy
 *
 * LEDS
 *  RED   Transmitting packet
 *  GREEN Received packet
 *  BLUE  Sync Node (also voting on sync)
*/

// Includes
#include "contiki.h"
#include "net/rime.h"
#include "random.h"
#include "dev/button-sensor.h"
#include "dev/leds.h"
#include "sys/rtimer.h"
#include "sys/ctimer.h"
#include "dev/cc2420.h"
#include "net/netstack.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Allows replacing/editing rtimer
#define rtimer_set_george rtimer_set // MUST PATCH RTIMER_SET!!!!!

#define TRUE 1
#define FALSE 0
typedef int bool;

// Compile Time Parameters
// 100ms (89 is desync code delay trim value, measured 22/07/15 DESYNC)
#define          rtPERIOD_DESYNC      ((RTIMER_SECOND/10)-89)
// 100ms (73 is sync code delay trim value, measured 22/07/15 SYNC)
#define          rtPERIOD_SYNC        ((RTIMER_SECOND/10)-73)
// number of channels in use for (wrapping with C+1)
#define          Chans          4
// Radio TX power: Between 1(min) and 31(max). CC2420_TXPOWER_MIN
```

```
60 #define          RADIOPWR            CC2420_TXPOWER_MAX
   // desync coupling
62 const double      alpha =          0.6;
   // sync coupling
64 const double      beta =           1.1150;
   // timing convergence threshold (b_thres) (30.5176 microsecond ticks)
66 rtimer_clock_t     tConvergedGuard =     5;
   // listening guard (t_guard) 2.5 ms each side = 164 (30.5176 us tick)
68 rtimer_clock_t     tGuard =         164;
   // consecutive T convergence before limimted listening
70 unsigned short     NC =             40;
   // consecutive T lost beacon packets, before unconverged
72 unsigned short     NL =             10;
   // consecutive periods without SYNC node before calling election.
74 unsigned short     NE =             10;
   // sync listening timer (128 = second). 6-8 works about right.
76 const clock_time_t    cListeningTicks =     8;
   // amount of time SYNC listens in own channel for collisins, etc.
78 const unsigned int    SyncListenNativeChan =  (RTIMER_SECOND/35);
   // limited listening timer set delay tweak parameter (best left alone)
80 const int          procoffset =      44;
   // 0=listen to next channel, 1=listen to random channel
82 #define          RANDOMLISTEN      0
   // Maximum value of the timer used.
84 #define          RTIMER_OVERFLOW     65535

86 // Timers
   struct rtimer        maintimer;
88 struct rtimer        syncradioofftimer;
   static struct ctimer  listentimer; // note, this is a ctimer
90
   // Time Variables
92 rtimer_clock_t      rtPERIOD = rtPERIOD_DESYNC;
   rtimer_clock_t      tFire = 0;
94 rtimer_clock_t      tNext = 0;
   rtimer_clock_t      tPrev = 0;
96 rtimer_clock_t      tPrevOld = 0;
   rtimer_clock_t      tNextFire = 0;
98 rtimer_clock_t      tFireOld = 0;
   rtimer_clock_t      tInterFiringOld = 0;
100 rtimer_clock_t      tInterFiring = 0;
   rtimer_clock_t      tDiff = 0;
102 rtimer_clock_t      tOffset = 0;
   rtimer_clock_t      tListen = 0;
```

```
104 rtimer_clock_t      tReceived = 0;

106 unsigned short      sJustFired = 0;
    unsigned short      sJustVoted = 0;
108 unsigned short      sConverged = 0;
    unsigned short      sLostBeacons = 0;
110 unsigned short      sHeardPrev = 0;
    unsigned short      sHeardNext = 0;
112 unsigned short      sHeardSync = 0;
    unsigned int        PacketsHeardDuringLI = 0;
114
    uint8_t           NodeChannelState;
116 uint8_t           ThisNodeType = 0;
    rimeaddr_t          ChannelSyncNode;
118 unsigned short      NEcount = 0;
    uint8_t           HighestElectionRoll = 0;
120 char                arrived[130];
    bool            isSelfJustVoted = 0;
122 bool            isVoteReceived = 0;
    uint16_t          channelTX = 20;
124 uint16_t          channelRX = 20;
    uint8_t         W = 0;
126 unsigned long long int  AFN = 0;
    uint8_t          Roll = 0;
128 uint8_t          rtimer_ret = 0;

130 // Contiki Application Process Definitions
    PROCESS(example_desync_process, "DT-SCS Light");
132 AUTOSTART_PROCESSES(&example_desync_process);

134 // Function Definitions
    static unsigned short NodeType(void);
136 static unsigned short IAmSync(void);
    static void FireCallback (void *ptr);
138 static void RadioPowered(int a);
    static void CheckConvergence(clock_time_t new, clock_time_t old);
140 static void broadcast_recv(struct broadcast_conn *c,
                const rimeaddr_t *from);
142 static unsigned short Converged(void);
    static unsigned short IAmSync(void);
144 static rtimer_clock_t rtimer_difference(rtimer_clock_t tCur,
                    rtimer_clock_t tPre);
146 static void ReceiverOnNextCallback (void *ptr);
    static void ReceiverOffNextCallback (void *ptr);
```

```
148 static void ReceiverOnPrevCallback (void *ptr);
    static void ReceiverOffPrevCallback (void *ptr);
150 static void processElectionVote(rimeaddr_t castID, uint8_t castVOTE);
    static void ListenCallback(void *ptr);
152 static void SyncProcess(rtimer_clock_t rxtime, rtimer_clock_t txtime);

154 // Receiver callback
    static const struct broadcast_callbacks
156                     broadcast_call = {broadcast_recv};
    static struct broadcast_conn broadcast;
158

    // Node Types
160 enum node_type_enum {
      SYNCNODE,
162   DESYNCNODE
    };
164

    // Channel States
166 enum chan_mode_enum{
      ELECTION,
168   CONVERGING,
      CONVERGED
170 };

172 // Beacon Packet Structure
    struct beacon_packet {
174   uint8_t      node_type;
      uint8_t      chan_mode;
176   rimeaddr_t     chan_sync;
      uint8_t      chan_nodes;
178   uint8_t      chan_native;
    };
180

    // Returns TRUE if channel converged, FALSE if not
182 static unsigned short
    Converged(void)
184 {
      return (sConverged > NC);
186 }

188 // Returns TRUE if self is SYNC and the channel isn't in election mode
    static unsigned short
190 IAmSync(void)
    {
```

```
192   // NOTE: (rimeaddr_cmp returns non-zero if the addresses are  same)
      return ( (rimeaddr_cmp(&ChannelSyncNode, &rimeaddr_node_addr) != 0)
194                   && (NodeChannelState != ELECTION) );
}
196
// Returns self's node type
198 static unsigned short
NodeType(void)
200 {
      if (IAmSync() == TRUE) {
202     return SYNCNODE;
      } else {
204     return DESYNCNODE;
      }
206 }

208 // calculate the number of nodes in each channel, based on the time
// difference between the previous node's firing, and ours.
210 //   Sometimes outputs erroneous values (noise).
uint8_t
212 NodesInChannel(rtimer_clock_t curr, rtimer_clock_t prev)
{
214   uint8_t Wt = 0;
      rtimer_clock_t diff = rtimer_difference(curr, prev);
216     float ans = (((float) rtPERIOD) / ((float) diff));
      Wt = (unsigned int)(ans + 0.5);
218   return Wt;
}
220
// calculate if we have converged based on our previous and current fire
222 // times. Must handle rtimer overflow every 2 seconds.
static void
224 CheckConvergence(clock_time_t new, clock_time_t old)
{
226   tInterFiringOld = tInterFiring;
      tInterFiring = rtimer_difference(new, old);
228
      if (tInterFiring > tInterFiringOld) {
230     tDiff = (tInterFiring - tInterFiringOld);
      } else {
232     tDiff = (tInterFiringOld - tInterFiring);
      }
234
      if (tDiff <= tConvergedGuard) {
```

```
236       sConverged++;
          if (sConverged > (NC+1)) {
238         sConverged = (NC+1);
          }
240     }

242     // not converged if we've lost more than NL beacons
        if (sLostBeacons > NL) {
244       sConverged = 0;
        }
246   }

248   // calculate our new fire time for DESYNC. Either update based on DESYNC
      // or just fire on next T if we've missed a packet.
250   // Must Handle overflows and missed tPrev/tNext.
      static void
252   calculateFireTimer(void)
      {
254     rtimer_clock_t  tempP = 0;
        rtimer_clock_t  tempC = 0;
256     rtimer_clock_t  tempN = 0;

258     // move tPrev to 0
        if (tNext > tPrev) {
260       tempC = tFire - tPrev;
          tempN = tNext - tPrev;
262       tempP = 0;
        } else { // rtimer overflowed
264       tempC = tFire + (RTIMER_OVERFLOW - tPrev) + 0;
          tempN = tNext + (RTIMER_OVERFLOW - tPrev) + 0;
266       tempP = 0;
        }
268

        // DESYNC update equation
270     tNextFire = rtPERIOD + (1.0-alpha) * tempC
                  + alpha * ( (tempP/2) + (tempN/2) );
272

        // rescale to abosolute tPrev
274     tNextFire += tPrev;

276     // If we heard both tNext and tPrev
        if (sHeardNext && sHeardPrev) {
278       // And the new fire time isn't crazy (more than 2 periods away
          // (this may want writing better)
```

```
280     if ((tNextFire-tFire) <= (2*rtPERIOD)) {
        if (!Converged()) {
282       // update when not converged
          rtimer_set_george(&maintimer, tNextFire, 1,
284         (rtimer_callback_t) FireCallback, NULL);
        } else {
286       // nothing here
        }
288     } else { // tNextFire More than 2 periods away, so fire on T
        tNextFire = tFire + rtPERIOD;
290     }
    } else { // missed either tNext or tPrev
292   tNextFire = tFire + rtPERIOD;
    }
294 }

296 // calculate our new fire time for SYNC. Either update based on SYNC
    // or just fire on next T if we've missed a packet.
298 // Must Handle overflows and missed rxtime/txtime.
    static void
300 SyncProcess (rtimer_clock_t rxtime, rtimer_clock_t txtime)
    {
302   rtimer_clock_t lastheard = rxtime;
    rtimer_clock_t lastfired = txtime;
304
    // calculate the phase difference between us and who we heard
306   // bit-logic requires correct size varibles to ignore offsets.
    unsigned int phi = lastfired - lastheard;
308
    // calculate new-phase-of-sync from the phi and the beta parameter.
310   unsigned long int npos = (((long double) beta) *
                  ((unsigned long int) phi));
312
    // cap 'new-phase-of-sync' at one period.
314   if (npos > rtPERIOD) {
      npos = rtPERIOD;
316   }

318   // calculate new fire time.
    tNextFire = lastfired + npos - phi + rtPERIOD + procoffset;
320 }

322 // The receiver interupt callback
    // run every time a packet is received on channel
```

```
324 static void
    broadcast_recv(struct broadcast_conn *c, const rimeaddr_t *from)
326 {
    // save reception time
328 rtimer_clock_t rRXtime = RTIMER_NOW();

330 // green debug led on
    leds_on(LEDS_GREEN);
332
    // random seed
334 random_init(rRXtime);

336 // copy RX BCN from packetbuf_dataptr() ==> r (type beacon_packet)
    struct beacon_packet r; // maybe put this global to avoid reinit?
338 memcpy(&r, packetbuf_dataptr(), sizeof(r));

340 // if we hear another unconverged mote, then we unconverge too
    rimeaddr_t ReportedSyncNode = r.chan_sync;
342 uint8_t    ReportedChanMode = r.chan_mode;

344 // wait for 1 period before claiming this node as SYNC node
    if(IAmSync() && !isSelfJustVoted) { // If I'm the SYNC
346   // SYNC node (me) must only react to other SYNC nodes on our CH
      if ((r.node_type == SYNCNODE) &&
348       (cc2420_get_channel() == channelRX)) { // if beacon packet
          // two steps ensures time correct but
350       // uses time to read the channel from radio PHY
          PacketsHeardDuringLI++;
352       tReceived = rRXtime;
      } else if(r.chan_mode != ELECTION && // if >1 sync per channel
354         !rimeaddr_cmp(&ReportedSyncNode, &rimeaddr_node_addr)
          && r.chan_native == channelTX) {
356       // if self is sync & another, conceed defeat if my ID<THEM
        if(rimeaddr_node_addr.u8[0] < ReportedSyncNode.u8[0]){
358       rimeaddr_copy(&ChannelSyncNode, &ReportedSyncNode);
        }
360   }
    } else { // DESYNC Code
362   // If channel is converged.
      if (Converged()) {
364     // turn the radio off after packet RX
        RadioPowered(0);
366   }
```

```
368       // If I was the last to transmit,
          // the next person to TX is my next neigbour
370       if (sJustFired > 0) {
            sJustFired = 0;
372         sHeardNext = 1;
            tNext = rRXtime;
374         calculateFireTimer();
            tPrevOld = tPrev;
376         // tPrev = tNext; // just 2 nodes in channel
            // (breaks limited listening, as difference = 0)
378       } else {
            // if I haven't just transmitted, always update the prev
380         sHeardPrev = 1;
            tPrevOld = tPrev;
382         tPrev = rRXtime;
          }
384
          // Does the received packet have a channel SYNC node (non NULL)
386       if (rimeaddr_cmp(&ReportedSyncNode, &rimeaddr_null) == 0) {
            sHeardSync = 1; //heard sync
388       }

390       // If we haven't had a Sync for NE, call election.
          if (NEcount >= NE) {
392         NodeChannelState = ELECTION;
            sJustVoted = 0;
394         NEcount = 0;
          }
396
          // If the received node is in election mode
398       if (ReportedChanMode == ELECTION) {
            processElectionVote(*from, ReportedSyncNode.u8[0]);
400         isVoteReceived = TRUE; // anticipate if collision occurs
            // If I haven't voted in the election, make me.
402         if (sJustVoted == 0) {
              NodeChannelState = ELECTION;
404         }
          }
406     }
        leds_off(LEDS_GREEN);
408 }

410 // calculate rtimer ticks between two times, irrespective of overflows
    static rtimer_clock_t
```

```
412 rtimer_difference(rtimer_clock_t a, rtimer_clock_t b)
{
414   rtimer_clock_t diff = 0;
   if (a < b) { // rtimer overflowed
416     diff = RTIMER_OVERFLOW - b + a;
   } else {
418     diff = a - b;
   }
420   return diff;
}
422
// wrapper to turn the radio on or off
424 static void
RadioPowered(int a)
426 {
   if (a > 0) {
428     NETSTACK_MAC.on();  // radio on
   } else {
430     NETSTACK_MAC.off(0);  // radio off
   }
432 }

434 // turn the radio on for the DESYNC Next Packet
static void
436 ReceiverOnNextCallback (void *ptr)
{
438   sHeardNext = 0; // clear heard variable
   RadioPowered(1); // radio on & wait twice guard time
440   rtimer_set_george(&maintimer, RTIMER_NOW() + (2*tGuard), 1,
          (rtimer_callback_t) ReceiverOffNextCallback, NULL);
442 }

444 // turn the radio off after the DESYNC Next Packet
static void
446 ReceiverOffNextCallback (void *ptr)
{
448   RadioPowered(0); // radio off
   // missed next packet reception (DESYNC Limited Listening)
450   if (sHeardNext == 0) {
     tPrevOld += rtPERIOD + 75; // +75 offset for timer delay
452     tNext += rtPERIOD + 75; // empircally found
     sLostBeacons++; // count lost beacon
454   }
```

```
456   sJustFired = 0;
      // RX guard ticks before expected.
458   rtimer_clock_t testtime =  rtPERIOD + tPrevOld - tGuard;
      rtimer_set_george(&maintimer, testtime, 1, (rtimer_callback_t)
460             ReceiverOnPrevCallback, NULL);
}
462
// turn the radio on for the DESYNC Previous Packet
464 static void
ReceiverOnPrevCallback (void *ptr)
466 {
      sHeardPrev = 0; // clear heard variable
468   RadioPowered(1); // radio on & wait twice guard time
      rtimer_set_george(&maintimer, RTIMER_NOW() + (2*tGuard), 1,
470           (rtimer_callback_t) ReceiverOffPrevCallback, NULL);
}
472
// turn the radio off after the DESYNC Previous Packet
474 static void
ReceiverOffPrevCallback (void *ptr)
476 {
      RadioPowered(0); // radio off
478   if (sHeardPrev == 0) { // missed previous packet reception
        tPrev += rtPERIOD + 75; // move on ~T (+75 empircally found)
480     sLostBeacons++;
      }
482
      // iff we missed either of the next or previous, fire on next period
484   if ( (sHeardPrev == 0) || (sHeardNext == 0) ) {
        tNextFire = tFire + rtPERIOD;
486   }

488   rtimer_set_george(&maintimer, tNextFire, 1,
            (rtimer_callback_t) FireCallback, NULL);
490 }

492 // Turn the receiver off after the sync listening period
static void
494 ListenCallback(void *ptr)
{
496   // Note current time
      rtimer_clock_t time_now = RTIMER_NOW();
498
      // Add extra receiving time for sync to listen in it's own channel
```

```
500   // for conflicting sync nodes, etc.
      cc2420_set_channel(channelTX); // avoid use a blocking while?
502   while(RTIMER_CLOCK_LT(RTIMER_NOW(),
          time_now + SyncListenNativeChan));
504   RadioPowered(0); // radio off
}
506
// Each time someone votes, we need to check if their vote
508 // is bigger than the previous max.
static void
510 processElectionVote(rimeaddr_t castID, uint8_t castVOTE)
{
512   //  If it is, we make them the SYNC.
      if (castVOTE > HighestElectionRoll) {
514     rimeaddr_copy(&ChannelSyncNode, &castID); // (dest, src)
        HighestElectionRoll = castVOTE;
516   } else if (castVOTE == HighestElectionRoll) { // are they equal
        // if they're equal, the higher u8[0] ID is chosen.
518     if (ChannelSyncNode.u8[0] >= castID.u8[0]) {
          rimeaddr_copy(&ChannelSyncNode, &castID); // (dest, src)
520     }
      }
522 }

524 // This code does the beacon firing for SYNC and DESYNC
static void
526 FireCallback (void *ptr)
{
528   struct beacon_packet b;
      int PacketBuffCopied = 0;
530
      leds_on(LEDS_RED);
532
      // save old fire time for examining offsets.
534   tFireOld = tFire;

536   RadioPowered(1); // turn the transceiver on

538   // check if we're converged - this updates NodeChannelState
      Converged();
540
      // create beacon packet, b.
542   b.node_type = NodeType(); // function returns DESYNCNODE or SYNCNODE
      b.chan_mode = NodeChannelState;
```

```
544   b.chan_sync = ChannelSyncNode;
      b.chan_native = channelTX;
546   b.chan_nodes = W;

548   sJustVoted = 0;
      isSelfJustVoted = FALSE;
550
      // if we're going to vote for an election, run this...
552   if (NodeChannelState == ELECTION) {
        //uint8_t Roll = (random_rand() & 0xFF);
554     rimeaddr_t RolledDice = rimeaddr_null;
        RolledDice.u8[0] = Roll;
556     RolledDice.u8[1] = Roll;

558     b.chan_mode = ELECTION;
        b.chan_sync = rimeaddr_node_addr;
560     rimeaddr_copy(&ChannelSyncNode, &RolledDice); // (dest, src)
        // process my own vode
562     processElectionVote(rimeaddr_node_addr, RolledDice.u8[0]);
        isSelfJustVoted = TRUE;
564
        // once we have voted, return to converging mode.
566     NodeChannelState = CONVERGING;
        sJustVoted = 1; // we have voted now, so stop doign it again.
568   }

570   // move the beacon over to the transmitter
      PacketBuffCopied = packetbuf_copyfrom(&b,
572                     sizeof(struct beacon_packet));

574   if (PacketBuffCopied != sizeof(struct beacon_packet)) {
        // We've not copied all of the data!
576     printf("**Transmit buffer copy failed! Copied %d of %d bytes\n",
            PacketBuffCopied, sizeof(struct beacon_packet));
578   }

580   cc2420_set_channel(channelTX);
      broadcast_send(&broadcast); // Transmit beacon.
582   tFire = RTIMER_NOW();
      AFN++; // update absoloute fire number!
584   W = NodesInChannel(tFire, tPrev); // update the nodes in each CH

586   // guess mext fire time, based on current plus period T.
      tNextFire = tFire + rtPERIOD;
```

```
588   sJustFired = 1;

590   // Check if we lost a packet - we must re-calculate p
      if ( (rtimer_difference(RTIMER_NOW(), tNext) > rtPERIOD) ||
592   (rtimer_difference(RTIMER_NOW(), tPrev) > rtPERIOD) ||
      (sHeardPrev == 0) || (sHeardNext == 0) ) {
594     if (NodeType() == DESYNCNODE) {
          sLostBeacons++;
596     }
      } else {
598     sLostBeacons = 0;
      }
600
      CheckConvergence(tFire, tFireOld);
602   Converged();

604     // wait for 1 period before self-claiming this node as SYNC
      if(IAmSync() && !isSelfJustVoted && isVoteReceived) {
606     // SYNC Code
        rtPERIOD = rtPERIOD_SYNC;
608     leds_on(LEDS_BLUE);
        if (PacketsHeardDuringLI == 0) {
610       tNextFire = tFire + rtPERIOD;
          //printf("no sync heard tFire=%u\n", tNextFire);
612     } else {
          SyncProcess(tReceived, tFire);
614     }
        //Randomise sync listening channel (if requested)
616     #if RANDOMLISTEN
          channelRX = 11+(random_rand()%Chans);
618     #else
          channelRX = channelTX + 1; // restore the RX CH to one up
620       if (channelRX > (11 + Chans - 1)) {
            channelRX = 11;
622       }
        #endif
624     //printf("Listening on channel %u\n", channelRX);
        PacketsHeardDuringLI = 0;
626     cc2420_set_channel(channelRX);
        // set timer to turn off receiver after cListeningTicks.
628     ctimer_set(&listentimer, cListeningTicks, ListenCallback, NULL);

630     // needs 16 bit values to ensure this works correctly
        if ((tNextFire - ((rtimer_clock_t) RTIMER_NOW())) >
```

```
632       (1.5*((double)rtPERIOD))) {
          printf("*** rtimer_set > 1.5 period away: %u -> %u\n",
634         tNextFire, RTIMER_NOW());
          //tNextFire -= rtPERIOD; // either this, or just fire on T
636     }
        rtimer_ret = rtimer_set_george(&maintimer, tNextFire, 1,
638       (rtimer_callback_t) FireCallback, NULL);
    } else {
640     // DESYNC Code
        rtPERIOD = rtPERIOD_DESYNC;
642     leds_off(LEDS_BLUE);
        channelRX = channelTX; // restore the RX channel
644     cc2420_set_channel(channelRX); // set RX on correct cjamme;
        if (Converged()) {
646       RadioPowered(0);
          rtimer_clock_t testtime =  rtPERIOD + tNext - tGuard;
648       // RX guard ticks before expected.
          rtimer_set_george(&maintimer, testtime, 1,
650         (rtimer_callback_t) ReceiverOnNextCallback, NULL);
        } else {
652       rtimer_set_george(&maintimer, tNextFire, 1,
            (rtimer_callback_t) FireCallback, NULL);
654     }

656     // did we hear a sync node this period (or, did any NBR?)
        if (sHeardSync == 0) {
658       NEcount++;
        } else {
660       NEcount = 0;
        }
662     sHeardSync = 0; // reset
    }
664

    leds_off(LEDS_RED);
666 }

668 // Static channel allocation to avoid runing balancing duing debug
    static void
670 setChannels (rimeaddr_t naddr)
    {
672   if ( (1 <= naddr.u8[0]) && (naddr.u8[0] <= 4) ) {
        channelTX = 11;
674     channelRX = 11;
        return;
```

```
676   }

678   if ( (5 <= naddr.u8[0]) && (naddr.u8[0] <= 8) ) {
        channelTX = 12;
680     channelRX = 12;
        return;
682   }

684   if ( (9 <= naddr.u8[0]) && (naddr.u8[0] <= 12) ) {
        channelTX = 13;
686     channelRX = 13;
        return;
688   }

690   if ( (13 <= naddr.u8[0]) && (naddr.u8[0] <= 16) ) {
        channelTX = 14;
692     channelRX = 14;
        return;
694   }

696   printf("Can't match Node: %d\n", naddr.u8[0]);
}
698
// Main program code
700 PROCESS_THREAD(example_desync_process, ev, data)
{
702   PROCESS_EXITHANDLER(broadcast_close(&broadcast);)
    PROCESS_BEGIN();
704
      // start the realtime scheduler
706   rtimer_init();

708   isSelfJustVoted = FALSE;
    isVoteReceived = FALSE;
710
    Roll = (random_rand() & 0xFF);
712
    // Set sync node to 0.0 (NULL)
714   ChannelSyncNode = rimeaddr_null;

716   // Set to DESYNC in CONVERGING state to begin
    NodeChannelState = CONVERGING;
718   ThisNodeType = DESYNCNODE;
```

```
720   // setup TX and RX channel based on node ID (full code has dynamic)
    setChannels(rimeaddr_node_addr);
722
    printf("DT-SCS Light - https://github.com/m1geo/DTSCS\n");
724   printf("George Smart, M1GEO. Elec Eng, UCL, UK.\n");
    printf("Node (%d.%d) on Channels %u/%u\n",
726     rimeaddr_node_addr.u8[0], rimeaddr_node_addr.u8[1],
      channelTX, channelRX);
728   printf("Compilation Timestamp %s - %s: %s\n",
      __TIME__, __DATE__, __FILE__);
730   printf("Period=%u, RTIMER_SECOND=%u (%u bytes)\n",
      rtPERIOD, RTIMER_SECOND, sizeof(rtimer_clock_t));
732
    random_init((rimeaddr_node_addr.u8[0]+rimeaddr_node_addr.u8[1]));
734
    broadcast_open(&broadcast, 129, &broadcast_call);
736
    cc2420_set_channel(channelRX);
738
    // start randomly through the period by calling the previous
740   //listening interval (otherwise network starts synced by Cooja)

742   tNextFire = (RTIMER_NOW() + (random_rand() % (rtPERIOD/10)));
    rtimer_set_george(&maintimer, tNextFire, 1,
744             (rtimer_callback_t) FireCallback, NULL);

746   PROCESS_END();
}
```

Code/DTSCS.c

7.3.2.2 Patching the Contiki-OS TelosB `rtimer` Library

To allow for successive updates of firing time, the `rtimer` Library needs to be patched. The file requiring patching is `contiki/core/sys/rtimer.c`. A patch file, `rtimer_patch.diff`, is provided which contains the modifications. This file *must* be patched *before* compiling, otherwise the protocol will not work correctly.

```
--- rtimer_original.c 2016-01-12 00:17:52.959177785 +0000
2 +++ rtimer_modded.c 2014-08-30 19:12:34.381996000 +0100
@@ -83,9 +83,9 @@
4     rtimer->time = time;
    next_rtimer = rtimer;
```

```
6
-   if(first == 1) {
8   +   //if(first == 1) {
        rtimer_arch_schedule(time);
10  -   }
    +   //}
12      return RTIMER_OK;
    }
```

Code/rtimer_patch.diff

Bibliography

[1] A. Tinka, T. Watteyne, and K. Pister, "A decentralized scheduling algorithm for time synchronized channel hopping," in *Ad Hoc Networks*. Springer, 2010, pp. 201–216.

[2] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, "IETF 6TSCH: Combining IPv6 connectivity with industrial performance," in *Proceedings of the 7th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'13)*. IEEE, 2013, pp. 541–546.

[3] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'11)*. New York, NY, USA: ACM, 2011, pp. 23:1–23:11.

[4] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08)*, 2008, pp. 53–63.

[5] D. Evans. The Internet of Things: How the Next Evolution of the Internet is Changing Everything. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[6] The Intel® IoT Platform. Architecture Specification White Paper: Internet of Things (IoT). [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/iot-platform-reference-architecture-paper.pdf

[7] ARM Ltd. ARM mbed IoT Device Platform. [Online]. Available: http://www.mbed.com/

[8] Microsoft Corporation. Azure IoT Suite. [Online]. Available: https://www.microsoft.com/en-gb/cloud-platform/internet-of-things-azure-iot-suite

[9] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, and P. Siano, "IoT-based smart cities: A survey," in *IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC'16)*, June 2016, pp. 1–6.

[10] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A survey of commercial frameworks for the Internet of Things," in *IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA'15)*, Sept 2015, pp. 1–8.

[11] V. C. Emeakaroha, N. Cafferkey, P. Healy, and J. P. Morrison, "A cloud-based IoT data gathering and processing platform," in *3rd International Conference on Future Internet of Things and Cloud (FiCloud'15)*, Aug 2015, pp. 50–57.

[12] G. Suciu, A. Vulpe, O. Fratu, and V. Suciu, "M2M remote telemetry and cloud IoT big data processing in viticulture," in *International Wireless Communications and Mobile Computing Conference (IWCMC'15)*, Aug 2015, pp. 1117–1121.

[13] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *IEEE International Conference on Services Computing (SCC'15),*, June 2015, pp. 285–292.

[14] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the Internet of Things using Big Data analytics," *Computer Networks*, vol. 101, pp. 63–80, 2016.

[15] Rolls-Royce PLC. Rolls-Royce takes TotalCare digital with Microsoft and Singapore Airlines. [Online]. Available: http://www.rolls-royce.com/media/press-releases/yr-2016/11-07-2016-rr-takes-totalcare-digital-with-microsoft-and-singapore-airlines.aspx

[16] S. Jain, V. K. N., A. Paventhan, V. K. Chinnaiyan, V. Arnachalam, and P. M., "Survey on smart grid technologies- smart metering, IoT and EMS," in *IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS'14)*, March 2014, pp. 1–6.

[17] S. H. Sutar, R. Koul, and R. Suryavanshi, "Integration of Smart Phone and IoT for development of smart public transportation system," in *International Conference on Internet of Things and Applications (IOTA'16)*, Jan 2016, pp. 73–78.

[18] S. Misbahuddin, J. A. Zubairi, A. Saggaf, J. Basuni, S. A-Wadany, and A. Al-Sofi, "IoT based dynamic road traffic management for smart cities," in *12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET'15)*, Dec 2015, pp. 1–5.

[19] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in *2012 10th International Conference on Frontiers of Information Technology*, Dec 2012, pp. 257–260.

[20] Crossbow Technology Inc., "Crossbow TelosB datasheet," 2013. [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf

[21] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, quarter 2010.

[22] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems (TECS'07)*, vol. 6, no. 4, p. 32, 2007.

[23] IEEE 802.11p, "IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std. 802.11p*, July 2010.

[24] IEEE 802.15.4-2011, "IEEE Standard for Local and Metropolitan Area Networks. Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std. 802.15.4*, 2011.

[25] A. Kunz, A. Prasad, K. Samdanis, S. Husain, and J. Song, "Enhanced 3GPP system for Machine Type Communications and Internet of Things," in *IEEE Conference on Standards for Communications and Networking (CSCN'15)*, Oct 2015, pp. 48–53.

[26] 3rd Generation Partnership Project. (2015) Workplan description: Release 13 analytical view. [Online]. Available: http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-13_description_20150917.zip

[27] L. Alliance. (2016) Lora® technology. [Online]. Available: https://www.lora-alliance.org/What-Is-LoRa/Technology

[28] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. Pister, "A realistic energy consumption model for TSCH networks," *IEEE Sensor Journal*, vol. 14, no. 2, pp. 482–489, 2013.

[29] Chipcon AS, "TI CC2420 datasheet," 2013. [Online]. Available: http://www.ti.com/product/cc2420

[30] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *Seventh IEEE International*

Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'96), vol. 2, Oct 1996, pp. 392–396 vol.2.

[31] D. Buranapanichkit and Y. Andreopoulos, "Distributed time-frequency division multiple access protocol for wireless sensor networks," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 440–443, Oct. 2012.

[32] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*. ACM, 2007, pp. 11–20.

[33] R. Pagliari, Y. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 4, pp. 564–575, May 2010.

[34] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1457–1486, Sept 2008.

[35] S. P. Beeby, M. J. Tudor, and N. White, "Energy harvesting vibration sources for microsystems applications," *Measurement Science and Technology*, vol. 17, no. 12, p. R175, 2006.

[36] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research B*, vol. 12, pp. 259–295, 2009.

[37] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power Sources for Wireless Sensor Networks," *Sensor Networks*, vol. 2920, pp. 1–17, 2004. [Online]. Available: http://www.springerlink.com/index/B0UTGM8AHNPHLL3L.pdf

[38] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *4th International Conference on Information Processing in Sensor Networks (IPSN'05)*, April 2005, pp. 457–462.

[39] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 443–461, Third 2011.

[40] M. Piñuela, P. D. Mitcheson, and S. Lucyszyn, "Ambient RF Energy Harvesting in Urban and Semi-Urban Environments," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 7, pp. 2715–2726, July 2013.

[41] H. Rahmani and A. Babakhani, "A fully integrated electromagnetic energy harvesting circuit with an on-chip antenna for biomedical implants in 180 nm SOI CMOS," in *2016 IEEE SENSORS*, Oct 2016, pp. 1–3.

[42] R. J. M. Vullers, R. v. Schaijk, H. J. Visser, J. Penders, and C. V. Hoof, "Energy Harvesting for Autonomous Wireless Sensor Networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, Spring 2010.

[43] S. Nabavi and L. Zhang, "MEMS piezoelectric energy harvester design and optimization based on Genetic Algorithm," in *2016 IEEE International Ultrasonics Symposium (IUS)*, Sept 2016, pp. 1–4.

[44] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design Considerations for Solar Energy Harvesting Wireless Embedded Systems," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147764

[45] H. Yu and H. Wu, "Design of power management ASIC for piezoelectric energy harvester," in *2016 IEEE SENSORS*, Oct 2016, pp. 1–3.

[46] S. M, K. S, H. L. R, and R. S, "Design of modified power conditioning circuit for piezoelectric vibration energy harvester," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, March 2016, pp. 2171–2176.

[47] Z. Wang, D. Tsonev, S. Videv, and H. Haas, "On the Design of a Solar-Panel Receiver for Optical Wireless Communications With Simultaneous Energy Harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1612–1623, Aug 2015.

[48] S. Carreon-Bautista, A. Eladawy, A. N. Mohieldin, and E. Sánchez-Sinencio, "Boost Converter With Dynamic Input Impedance Matching for Energy Harvesting With Multi-Array Thermoelectric Generators," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5345–5353, Oct 2014.

[49] H. Besbes, G. Smart, D. Buranapanichkit, C. Kloukinas, and Y. Andreopoulos, "Analytic conditions for energy neutrality in uniformly-formed wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 4916–4931, 2013.

[50] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.

[51] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1326–1336, Apr. 2010.

[52] Y. Andreopoulos and M. Van der Schaar, "Adaptive linear prediction for resource estimation of video decoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 6, pp. 751–764, 2007.

[53] N. Kontorinis, Y. Andreopoulos, and M. Van der Schaar, "Statistical framework for video decoding complexity modeling and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1000–1013, 2009.

[54] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design Test of Computers*, vol. 18, no. 2, pp. 62–74, Mar 2001.

[55] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets'07)*. New York, NY, USA: ACM, 2007, pp. 28–32. [Online]. Available: http://doi.acm.org/10.1145/1278972.1278979

[56] H. Ritter, J. Schiller, T. Voigt, A. Dunkels, and J. Alonso, "Experimental evaluation of lifetime bounds for wireless sensor networks," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, Jan 2005, pp. 25–32.

[57] N. A. Pantazis, D. J. Vergados, D. D. Vergados, and C. Douligeris, "Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling," *Ad Hoc Networks*, vol. 7, no. 2, pp. 322–343, 2009.

[58] F. Salvadori, M. de Campos, P. S. Sausen, R. F. de Camargo, C. Gehrke, C. Rech, M. A. Spohn, and A. C. Oliveira, "Monitoring in industrial systems using wireless sensor network with dynamic power management," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3104–3111, 2009.

[59] P. N. Whatmough, G. Smart, S. Das, Y. Andreopoulos, and D. M. Bull, "A 0.6V all-digital body-coupled wakeup transceiver for IoT applications," in *2015 Symposium on VLSI Circuits (VLSIC'15)*, June 2015, pp. C98–C99.

[60] C. He, M. E. Kiziroglou, D. C. Yates, and E. M. Yeatman, "A MEMS self-powered sensor and RF transmission platform for WSN nodes," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3437–3445, Dec. 2011.

[61] S. Chalasani and J. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE Southeastcon, 2008*, Apr. 2008, pp. 442 –447.

[62] A. Koubaa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for IEEE 802.15.4/zigbee cluster-tree wireless sensor networks," in *19th Euromicro Conference on Real-Time Systems (ECRTS'07)*, july 2007, pp. 125–135.

[63] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proceedings of the 23rd IEEE International Performance Computing and Communications Conference (IPCCC'14)*. IEEE, 2004, pp. 701–706.

[64] B. Zhang, R. Simon, and H. Aydin, "Energy management for time-critical energy harvesting wireless sensor networks," in *Lecture Notes in Computer Science: Stabilization, Safety, and Security of Distributed Systems*, vol. 6366, 2010, pp. 236–251.

[65] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "Energy management in wireless sensor networks with energy-hungry sensors," *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 2, pp. 16–23, 2009.

[66] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, 2000.

[67] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 1180–1189, 2012.

[68] N. Abramson, "THE ALOHA SYSTEM: Another Alternative for Computer Communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference (AFIPS'70 Fall)*. New York, NY, USA: ACM, 1970, pp. 281–285. [Online]. Available: http://doi.acm.org/10.1145/1478462.1478502

[69] A. S. Tanenbaum and D. J. Wetherall, *Computer networks (4th edition)*. Prentice-Hall, 2013.

[70] L. G. Roberts, "ALOHA Packet System With and Without Slots and Capture," *SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, Apr. 1975. [Online]. Available: http://doi.acm.org/10.1145/1024916.1024920

[71] A. Dunkels, "The ContikiMAC radio duty cycling protocol," SICS, Tech. Rep. ISSN 1100-3154, 2011.

[72] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for duty-cycled Wireless Sensor Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 307–320. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182838

[73] R. R. Parry, "AX.25 [data link layer protocol for packet radio networks]," *IEEE Potentials*, vol. 16, no. 3, pp. 14–16, Aug 1997.

[74] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, 2012.

[75] H. Takagi and L. Kleinrock, "Throughput analysis for persistent CSMA systems," *IEEE Transactions on Communications*, vol. 33, no. 7, p. 627–638, July 1985.

[76] S. Abukharis and T. O'Farrell, "A new adaptive differentiated p-persistent CSMA protocol that reduces the effect of the transmission errors on the system throughput performance," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC'11)*, July 2011, pp. 2181–2185.

[77] M. Y. Darus, A. Kamarudin, N. Awang, and F. H. M. Ali, "Analysis performance on contention-based MAC protocols in MANETs," in *Fourth World Congress on Information and Communication Technologies (WICT'14)*, Dec 2014, pp. 303–307.

[78] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[79] S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. Van der Perre, F. Catthoor, I. Moerman, and A. Bahai, "Distributed cognitive coexistence of 802.15.4 with 802.11," in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, 2006, pp. 1–5.

[80] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, "OpenWSN: a standards-based low-power wireless development environment,"

Transactions on Emerging Telecommunications Technologies, vol. 23, no. 5, pp. 480–493, 2012.

[81] IEEE 802.15.4e-2012, "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*, April 2012.

[82] P. Zand, S. Chatterjea, K. Das, and P. Havinga, "Wireless industrial monitoring and control networks: The journey so far and the road ahead," *Journal of Sensor and Actuator Networks*, vol. 1, no. 2, pp. 123–152, 2012.

[83] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08)*, 2008, pp. 70–81.

[84] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.

[85] J. Hwang, T. Kim, J. So, and H. Lim, "A receiver-centric multi-channel MAC protocol for wireless networks," *Computer Communications*, vol. 36, no. 4, pp. 431–444, 2012.

[86] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys'08)*. New York, NY, USA: ACM, 2008, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460414

[87] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *The 30th IEEE International Conference on Computer Communications (INFOCOM'11)*, 2011, pp. 1305–1313.

[88] M. J. Miller and N. H. Vaidya, "A MAC protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228–242, 2005.

[89] M. Ali, T. Suleman, and Z. A. Uzmi, "MMAC: A mobility-adaptive, collision-free MAC protocol for wireless sensor networks," in *IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*. IEEE, 2005, pp. 401–407.

[90] T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in *International Conference on Broadband Communications, Networks and Systems (BROADNETS'06)*. IEEE, 2006, pp. 1–10.

[91] J. Borms, K. Steenhaut, and B. Lemmens, "Low-overhead dynamic multi-channel MAC for wireless sensor networks," in *Wireless Sensor Networks*. Springer, 2010, pp. 81–96.

[92] K. Pister and L. Doherty, "TSMP: Time synchronized mesh protocol," *IASTED Distributed Sensor Networks*, pp. 391–398, 2008.

[93] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'08)*. IEEE, 2008, pp. 377–386.

[94] IEEE 1609.4-2016, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation," *IEEE Std. 1609.4*, 2016.

[95] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensing platforms," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 882–901, 2009.

[96] A. Vinel, E. Belyaev, K. Egiazarian, and Y. Koucheryavy, "An overtaking assistance system based on joint beaconing and real-time video transmission," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 2319–2329, 2012.

[97] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457–469, 2011.

[98] N. Alam, A. T. Balaie, and A. G. Dempster, "A DSRC-based traffic flow monitoring and lane detection system," in *Proceedings of the 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2011, pp. 1–5.

[99] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.

[100] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y.-Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[101] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *International Journal on Advanced Robotic Systems*, vol. 10, no. 28, 2013.

[102] L. Merino, A. Gilbert, J. Capitán, R. Bowden, J. Illingworth, and A. Ollero, "Data fusion in ubiquitous networked robot systems for urban services," *Annales des Télécommunications*, vol. 67, no. 7-8, pp. 355–375, 2012.

[103] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: From mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, 2011.

[104] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.

[105] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[106] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets'07)*. ACM, 2007, pp. 78–82.

[107] N. Deligiannis, F. Verbist, J. Slowack, R. v. d. Walle, P. Schelkens, and A. Munteanu, "Progressively refined Wyner-Ziv video coding for visual sensors," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, p. 21, 2014.

[108] J. Xu, Y. Andreopoulos, Y. Xiao, and M. Van der Schaar, "Non-Stationary Resource Allocation Policies for Delay-Constrained Video Streaming: Application to Video over Internet-of-Things-Enabled Networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 782–794, 2014.

[109] Y.-Y. Shih, W.-H. Chung, P.-C. Hsiu, and A.-C. Pang, "A mobility-aware node deployment and tree construction framework for ZigBee wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2763–2779, 2013.

[110] N. Deligiannis, F. Verbist, J. Barbarien, J. Slowack, R. Van de Walle, P. Schelkens, and A. Munteanu, "Distributed coding of endoscopic video," in *IEEE International Conference on Image Processing (ICIP'11)*. IEEE, 2011, pp. 1813–1816.

[111] G. Smart, N. Deligiannis, Y. Andreopoulos, R. Surace, V. Loscri, and G. Fortino, "Decentralized time-synchronized channel swapping for wireless sensor networks," in *11th European Conference on Wireless Sensor Networks (EWSN'14), poster presentation*, 2014.

[112] D. Buranapanichkit, N. Deligiannis, and Y. Andreopoulos, "Convergence of desynchronization primitives in wireless sensor networks: A stochastic modeling approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 221–233, 2015.

[113] N. Deligiannis, J. F. Mota, G. Smart, and Y. Andreopoulos, "Decentralized multichannel medium access control: Viewing desynchronization as a convex optimization method," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN'15)*. ACM, 2015, pp. 13–24.

[114] N. Deligiannis, J. F. C. Mota, G. Smart, and Y. Andreopoulos, "Fast desynchronization for decentralized multichannel medium access control," *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3336–3349, Sept 2015.

[115] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[116] Arduino. The Arduino/Genuino UNO. [Online]. Available: http://www.arduino.cc/en/Main/ArduinoBoardUno

[117] MIDÉ, "Volture piezoelectric energy harvesters datasheet." [Online]. Available: http://www.mide.com/pdfs/Volture_Datasheet_001.pdf

[118] A. Koubaa, M. Alves, M. Attia, and A. Van Nieuwenhuyse, "Collision-free beacon scheduling mechanisms for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks," in *Proceedings of the 7th International Workshop on Applied and Services in Wireless Networks. (ASWN'07)*, 2007.

[119] J. Degesys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'08)*, 2008, pp. 129–138.

[120] A. Motskin, T. Roughgarden, P. Skraba, and L. Guibas, "Lightweight coloring and desynchronization for networks," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM'09)*. IEEE, 2009, pp. 2383–2391.

[121] T. Wu and S. Biswas, "Minimizing inter-cluster interference by self-reorganizing MAC allocation in sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 691–703, Oct. 2007.

[122] G. Haigang, L. Ming, W. Xiaomin, C. Lijun, and X. Li, "An interference free cluster-based TDMA protocol for wireless sensor networks," *Wireless Algorithms, Systems, and Applications*, vol. 4138, pp. 217–227, 2006.

[123] X. Zhuang, Y. Yang, and W. Ding, "A TDMA-based Protocol and Implementation for Avoiding Inter-Cluster Interference of Wireless Sensor Network," in *IEEE International Conference on Industrial Technology (ICIT'09)*, 2009, pp. 1–6.

[124] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *4th Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'07)*, Jun. 2007, pp. 21 –30.

[125] J. Kho, A. Rogers, and N. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, p. 19, 2009.

[126] A. McCree, K. Brady, and T. Quatieri, "Multisensor very lowbit rate speech coding using segment quantization," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08)*. IEEE, 2008, pp. 3997–4000.

[127] M. Pursley and L. Davisson, "Variable rate coding for nonergodic sources and classes of ergodic sources subject to a fidelity constraint," *IEEE Transactions on Information Theory*, vol. 22, no. 3, pp. 324–337, 1976.

[128] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: a multi-tier camera sensor network," in *13th Annual ACM International Conference on Multimedia*. ACM, 2005, pp. 229–238.

[129] A. Rowe, D. Goel, and R. Rajkumar, "Firefly mosaic: A vision-enabled wireless sensor networking system," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS'07)*. IEEE, 2007, pp. 459–468.

[130] M. Tagliasacchi, S. Tubaro, and A. Sarti, "On the modeling of motion in Wyner-Ziv video coding," in *IEEE International Conference on Image Processing (ICIP'06)*. IEEE, 2006, pp. 593–596.

[131] A. Redondi, M. Cesana, and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors," in *MMSP*, 2012, pp. 124–129.

[132] G. WernerAllen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *7th Symposium on Operating Systems Design and Implementation (OSDI'06)*. ACM, 2006, pp. 381–396.

[133] D. Palma, L. Bencini, G. Collodi, G. Manes, F. Chiti, R. Fantacci, and A. Manes, "Distributed monitoring systems for agriculture based on wireless sensor network technology," *International Journal on Advances in Networks and Services*, vol. 3, 2010.

[134] A. Redondi, M. Chirico, L. Borsani, M. Cesana, and M. Tagliasacchi, "An integrated system based on wireless sensor networks for patient monitoring, localization and tracking," *Ad Hoc Networks*, vol. 11, no. 1, pp. 39–53, 2013.

[135] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[136] B. Foo, Y. Andreopoulos, and M. Van der Schaar, "Analytical rate-distortion-complexity modeling of wavelet-based video coders," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 797–815, Feb. 2008.

[137] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, Mar. 1995.

[138] K. Park, G. Kim, and M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proceedings of the 1996 International Conference on Network Protocols (ICNP'96)*. IEEE, 1996, pp. 171–180.

[139] M. Dai, Y. Zhang, and D. Loguinov, "A unified traffic model for MPEG-4 and H.264 video traces," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 1010–1023, May 2009.

[140] Y. Andreopoulos and I. Patras, "Incremental refinement of image salient-point detection," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1685–1699, Sept. 2008.

[141] A. Papoulis, *Probability & statistics (3rd edition)*. Prentice-Hall, 1990.

[142] W. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *Applied Statistics*, vol. 41, no. 2, pp. 337–348, 1992.

[143] J. Notay and G. Safdar, "A wireless sensor network based structural health monitoring system for an airplane," in *Proceedings of the 17th IEEE International Conference on Automation and Computing (ICAC'11)*. IEEE, 2011, pp. 240–245.

[144] A. Patel, J. Degesys, and R. Nagpal, "Desynchronization: The theory of self-organizing algorithms for round-robin scheduling," *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, july 2007.

[145] C.-M. Lien, S.-H. Chang, C.-S. Chang, and D.-S. Lee, "Anchored desynchronization," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12)*, 2012, pp. 2966–2970.

[146] J. Klinglmayr and C. Bettstetter, "Self-organizing synchronization with inhibitory-couples oscillaotrs: convergence and robustness," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, no. 3, Sep. 2012.

[147] C. A. Boano, T. Voigt, C. Noda, K. Romer, and M. Zúñiga, "Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *10th International Conference on Information Processing in Sensor Networks (IPSN'11)*. IEEE, 2011, pp. 175–186.

[148] B. O. Ayinde and A. Y. Barnawi, "Differential evolution based deployment of wireless sensor networks," in *IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA'14)*, Nov 2014, pp. 131–137.

[149] A. Redondi, D. Buranapanichkit, M. Cesana, M. Tagliasacchi, and Y. Andreopoulos, "Energy Consumption of Visual Sensor Networks: Impact of Spatio-Temporal Coverage," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2117–2131, Dec 2014.

[150] F. Renna, J. Doyle, V. Giotsas, and Y. Andreopoulos, "Media Query Processing For The Internet-of-Things: Coupling Of Device Energy Consumption And Cloud Infrastructure Billing," *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1, 2016.

[151] V. G. Kulkarni, *Modeling and analysis of stochastic systems*. CRC Press, 1995, vol. 36.