Boano Carlo Alberto, Dott. Dott. mag. MSc

Dependable Wireless Sensor Networks

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften (Dr. techn.)

eingereicht an der
Technischen Universität Graz

Betreuer
Univ.-Prof. Dipl.-Inform. Dr. sc. ETH Kay Uwe Römer

Institut für Technische Informatik

Graz, im Oktober 2014

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

I declare that I have authored this Thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral dissertation.

.... (Datum / Date)

.... (Unterschrift / Signature)

To my family
Alla mia famiglia

Acknowledgements

This doctoral thesis is the result of a long, rewarding journey, which began five years ago on the shores of the Baltic Sea, and is now completing in the shadow of the Alps. During these years I had the privilege to work with amazing people, who gave me the possibility to grow both professionally and personally, and it's anything but easy to thank all them in a few lines.

I begin by thanking my advisor, Prof. Kay Römer, for believing in me and for his dedicated and patient guidance. I owe most of my academic success to him, and working in his group was a truly remarkable experience. I am proud to have had as advisor one of the most acclaimed and remarkable researchers in the networked embedded field, and I am deeply indebted for all the help and constructive comments received throughout these years: from the feedback on my experimental results and the technical discussions over lunch, to the valorization of my research contributions and the correction of drafts the night before the deadline.

I would also like to thank Prof. Koen Langendoen, who kindly agreed to be the external examiner of my thesis: his feedback in the final stage of the writing process has been extremely valuable.

A special thanks goes to Prof. Thiemo Voigt and the whole group at SICS for the deep interest in sensor networks research that they aroused in me while carrying out my Master thesis in Stockholm. Thiemo not only introduced me to the (unknown) world of research, but also shared me his boundless motivation: looking backwards, the decision of being a researcher and starting this journey actually matured while working in his group.

Invaluable have also been the brilliant researchers I had the pleasure to work with during these years: James Brown, Zhitao He, Chamath Keppitiyagama, Luca Mottola, Claro Noda, Utz Roedig, Nicolas Tsiftes, Thiemo Voigt, Hjalmar Wennerström, and Marco Zúñiga. I truly have unforgettable memories of some of us working towards 6 AM paper deadlines, and of the great motivation and atmosphere that you created around my research. I would like to especially thank Marco, who has always been ready to discuss with me ideas and analyse results, no matter if it was late evening or weekend.

I would also like to thank all the people who helped me spiritually and morally during this memorable phase of my life, starting with Matteo Lasagni with whom I shared a large part of this long journey.

From an unknown researcher met at a conference on the other side of the world, Matteo has actually became a wonderful friend during the last four years. I am really glad that he was there, whenever I needed to talk or distract myself from work, and whenever I required some technical help. I also treasured the friendship of Alberto, who has always been very supportive despite the distance.

I would then like to thank all my friends in Lübeck for making my four-years stay in the charming Hanseatic city so special. This parenthesis of my life will remain forever in my heart, together with countless unforgettable moments shared together. A deep thanks also goes to my colleagues at ITI, especially Richard, Felix Jonathan, Grigore, Cuong, Patrick, and obviously Matteo.

Matteo and Felix Jonathan have actually been the ones from which I received most technical help for my research, and I am very happy that they have also joined the new group in Graz and kept being very supportive. Besides them, I would also like to thank all the people in Graz that had an impact on my professional and personal life, from my other colleagues at the Institute to the friends outside the University with whom I shared good moments and laughter.

A heartfelt gratitude goes to my beautiful girlfriend Nóra for all her love, immeasurable patience and invaluable support. No-one would have been able to devote me as much time and attention, silently agreeing to share me between my beloved laptop and wireless sensors. Thank you for accepting my passion for research and for doing your best to draw my attention away from work and to make me smile in the most difficult moments.

I finally want to dedicate this dissertation to my family: my parents and my sisters. Thank you for supporting my decision to study abroad unconditionally: I am perfectly aware that having me far away was burdensome, and dedicating you my achievements is the least I can do. Thank you from the deepest of my heart, for all your love and support, and for being so close although so far away.

Graz, 20.10.2014
Carlo Alberto Boano

The research leading to this dissertation has been partially supported by the European Commission with contracts FP7-ICT-224282 (GINSENG, Performance Control in Wireless Sensor Networks), FP7-2007-2-224053 (CONET, the Cooperating Objects Network of Excellence), and INFSO-ICT-317826 (RELYonIT, Research by Experimentation for Dependability on the Internet of Things), as well as by the DFG-funded Cluster of Excellence 306/1 “Inflammation at Interfaces”.

Ringraziamenti

Questa tesi di dottorato è il coronamento di un lungo e gratificante percorso di studi, iniziato cinque anni fa sulle rive del Mar Baltico e conclusosi ora all'ombra delle Alpi. Durante questi anni, ho avuto il privilegio di lavorare con persone fantastiche, che mi hanno dato la possibilità di crescere sia professionalmente che umanamente, e non è sicuramente semplice ricordare e ringraziare ognuno di loro in queste poche righe.

Inizio con il mio relatore, il Prof. Kay Römer, per aver creduto in me e per la sua scrupolosa e paziente guida. A Lui devo la maggior parte del mio successo accademico, ed è stato davvero un onore poter lavorare al suo fianco. Sono ovviamente lusingato di aver avuto come supervisore uno dei più acclamati esperti del settore, al quale sono profondamente riconoscente per tutto l'aiuto fornitomi durante questi anni: per avermi saputo ascoltare ed interpretare le mie esigenze, valorizzando il mio lavoro e offrendomi sempre la massima disponibilità.

Un ringraziamento particolare va anche al Prof. Koen Langendoen, che ha gentilmente accettato di essere l'esaminatore esterno della mia tesi: i suoi commenti nella fase finale della stesura sono stati molto costruttivi.

Intendo poi ringraziare il Prof. Thiemo Voigt e tutto il suo gruppo al SICS per aver saputo suscitare in me l'interesse nelle reti di sensori: la decisione di perseguire il dottorato di ricerca è maturata in me concretamente proprio mentre lavoravo nel loro straordinario gruppo. Senza gli incoraggiamenti e la motivazione che ha saputo infondere in me Thiemo, non avrei sicuramente mai raggiunto questo significativo traguardo.

Preziosissimi anche i brillanti ricercatori con cui ho avuto il piacere di lavorare durante questi anni: James Brown, Zhitao He, Chamath Keppitiyagama, Luca Mottola, Claro Noda, Utz Roedig, Nicolas Tsiftes, Thiemo Voigt, Hjalmar Wennerström, e Marco Zúñiga, che hanno speso parte del loro tempo per leggere e discutere con me le bozze del lavoro e per aver facilitato le mie ricerche. Vorrei ringraziare in particolare Marco, sempre disponibile indipendentemente dal giorno della settimana e dall'ora.

———

Vorrei inoltre ringraziare le persone che mi hanno aiutato spiritualmente e moralmente, dandomi il loro supporto e pazientando durante questa indimenticabile fase della mia vita, iniziando con Matteo Lasagni con il quale ho condiviso gran parte di questo lungo viaggio.

Da sconosciuto ricercatore incontrato in una conferenza dall'altra parte del mondo, Matteo è diventato a tutti gli effetti, durante gli ultimi quattro anni, un carissimo e fantastico

Amico. Ci tengo inoltre a ringraziare Alberto per il suo sostegno e per essere un buon amico anche a distanza e tutti gli amici e la comunità italiana di Lubecca per aver reso i quattro anni di permanenza nell'incantevole città Anseatica così speciali. Rimarranno sicuramente per sempre nel mio cuore, insieme ad un'infinità di momenti indimenticabili condivisi insieme.

Un immenso grazie va anche ai miei colleghi all'ITI, specialmente Richard, Felix Jonathan, Grigore, Cuong, Patrick, oltre ovviamente Matteo. Matteo e Felix Jonathan in primis sono stati quelli con cui ho lavorato più a stretto contatto e da cui ho ricevuto un cospicuo supporto tecnico, e sono molto felice di averli ritrovati a Graz, dove hanno continuato ad essermi di grande sostegno. Oltre a loro, voglio anche ringraziare tutte le altre persone che hanno avuto un'incidenza sulla mia vita personale ed accademica a Graz, dai miei colleghi d'Istituto, agli amici al di fuori dell'Università, con cui ho condiviso risate e bei momenti.

Proseguendo i ringraziamenti alle persone a me più care un ringraziamento particolare va a Nóra per tutto il suo affetto, la sua incommensurabile pazienza ed impagabile sostegno. Nessuno avrebbe saputo dedicarmi altrettanto tempo ed attenzione, accettando silenziosamente di condividermi tra il mio amato portatile e le mie decine di sensori wireless. Grazie per aver accettato la mia passione per la ricerca e per aver fatto del tuo meglio per farmi distrarre e sorridere nei momenti più difficili.

Voglio infine dedicare questa dissertazione alla mia famiglia: ai miei genitori ed alle mie sorelle. Grazie per aver sostenuto incondizionatamente la mia decisione di studiare all'estero: sono perfettamente cosciente che avermi lontano non sia stato facile, e dedicarvi i miei risultati è il minimo che io possa fare. Grazie dal più profondo del cuore per tutto il vostro amore e tutto il vostro appoggio, e per essermi stati così vicini, nonostante la distanza.

Graz, 20.10.2014
Carlo Alberto Boano

La ricerca alla base di questa tesi di dottorato è stata parzialmente finanziata dalla Commissione Europea con contratti FP7-ICT-224282 (GINSENG, Performance Control in Wireless Sensor Networks), FP7-2007-2-224053 (CONET, the Cooperating Objects Network of Excellence), e INFSO-ICT-317826 (RELYonIT, Research by Experimentation for Dependability on the Internet of Things), e dall'Excellence Cluster 306/1 "Inflammation at Interfaces" sponsorizzato dalla German Research Foundation (DFG).

Abstract

The ability to accurately monitor real-world phenomena by embedding tiny wireless sensors into the environment has revolutionized sensing in several application domains. Wireless sensor networks have been installed in civil, scientific, military, agricultural, and industrial settings during the past two decades, and are now ready to pervade our daily life. Indeed, they are becoming an integral part of the Internet of Things that will provide on-line access to the state of things and places in the years to come.

With potentially millions of users ready to use a plethora of attractive services and applications, the operations of wireless sensor networks must become highly dependable: sensed values need to be delivered reliably and timely, and the availability of a network should reach time-scales in the order of years. Failure to meet these requirements may result in high costs, insufficient user satisfaction, and physical damage to people or things.

The environment in which sensor nodes are embedded often plays a critical role for their performance, especially with respect to the energy efficiency and reliability of their communications. Temperature variations can cause loss of synchronization and degradation of the wireless link quality. Radio interference from surrounding wireless devices and electrical appliances may impair packet reception, reduce throughput and lead to high latencies. This vulnerability to the surrounding environment affects the capability of sensor networks to meet application-specific dependability requirements. Solving this problem is a fundamental step required to bring the visions and promises of the Internet of Things to reality.

Our thesis is that through a deeper understanding of the environmental impact, it is possible to increase the dependability of wireless sensor networks by designing more reliable and energy-efficient communication protocols. We specifically address two environmental parameters, namely temperature and radio interference, and support this thesis by first developing TempLab and JamLab, low-cost experimental facilities that allow to study, respectively, the impact of temperature and radio interference on sensornet hardware and protocols in a reproducible manner.

We leverage TempLab and JamLab to identify and highlight the limitations of state-of-the-art communication protocols in the presence of varying environmental conditions. We show that existing communication protocols are strongly affected by on-board temperature variations commonly found in outdoor deployments, and that traditional packet-based handshakes are not suitable in environments rich of radio interference.

The inadequacy of traditional communication protocols calls for the design and implementation of environment-aware protocols that increase the reliability and efficiency of wireless sensor networks deployed in harsh environments. Towards this goal, we derive models that capture the environmental impact with an accuracy that is sufficiently high

to parametrize protocols such that specific dependability requirements are met. Specifically, we characterize the signal strength attenuation as a function of temperature, and derive a platform-independent analytical model that is used to design temperature-aware protocols. We further model the statistical distribution of radio interference and use this information to design interference-aware protocols.

Building upon the devised environmental models, we dynamically adapt the clear channel assessment threshold to temperature changes in order to mitigate the impact that temperature variations have on carrier sense multiple access protocols. We further improve the reliability of an existing duty-cycled MAC protocol in the presence of radio interference and develop JAG, a protocol that uses a jamming sequence of configurable size to make sure that two neighbouring nodes agree on a given piece of information in radio environments with high interference.

Zusammenfassung

Drahtlose Sensornetze wurden in den vergangenen zwei Jahrzehnten vermehrt im zivilen, wissenschaftlichen, militärischen, landwirtschaftlichen und industriellen Bereich eingesetzt und beginnen unser tägliches Leben zu durchdringen. Diese kleinen vernetzten Geräte sind insbesondere auch ein zentraler Bestandteil des Internets der Dinge, das in naher Zukunft einen Echtzeitzugriff auf den Zustand von Dingen und Orten erlauben wird.

Mit Millionen potentieller Nutzer attraktiver Dienste und Anwendungen, müssen drahtlose Sensornetze eine sehr hohe Zuverlässigkeit aufweisen: Erfasste Werte sollten zuverlässig und rechtzeitig geliefert werden und die Netze sollten über eine Lebensdauer von Jahren verfügbar sein. Die Nichterfüllung dieser Anforderungen kann zu hohen Kosten, mangelnder Benutzerzufriedenheit, und Verletzungen oder Sachschäden führen.

Die Umgebung in welcher Sensornetze eingebettet werden spielt für die Leistungsfähigkeit häufig eine entscheidende Rolle, insbesondere hinsichtlich der Energieeffizienz und der Zuverlässigkeit der Kommunikation. Temperaturschwankungen können zu einem Synchronisationsverlust und Verbindungsproblemen führen. Funkstörungen durch andere Anwendungen und elektrische Geräte beeinträchtigen die Übertragung beispielsweise durch eine Verringerung des Datendurchsatzes und eine hohe Verzögerung. Diese Anfälligkeit gegenüber Umgebungsparametern erschwert die Erfüllung anwendungsspezifischer Zuverlässigkeitsanforderungen. Die Lösung dieses Problems stellt eine grundlegende Voraussetzung für die Realisierung der Vision des Internet der Dinge dar.

Unsere Hypothese ist, dass mit einem tieferen Verständnis der Umweltauswirkungen die Zuverlässigkeit von drahtlosen Sensornetzen durch die Erstellung verlässlicher und verfügbarer Kommunikationsprotokolle erhöht werden kann. Wir betrachten dabei insbesondere den Einfluss von Temperatur und Funkstörungen und entwickeln zur Prüfung unsere Hypothese die kostengünstigen Testumgebungen TempLab und JamLab, die erlauben die Auswirkung von Umgebungseinflüssen auf Sensornetze reproduzierbar zu untersuchen.

Wir nutzen TempLab und JamLab um Schwächen bestehender Kommunikationsprotokolle unter veränderlichen Umgebungsbedingungen zu untersuchen. Es zeigt sich, dass aktuelle Protokolle stark von Temperaturschwankungen, wie man sie häufig bei einem Außeneinsatz vorfindet, beeinflusst werden und dass traditionelle paketbasierte Handshake-Verfahren in Umgebungen mit starken Funkstörungen nicht geeignet sind.

Die Unzulänglichkeit der traditionellen Protokolle erfordert die Verbesserung der Zuverlässigkeit von drahtlosen Sensornetzen in anspruchsvollen Umgebungen durch den Entwurf sich den Umgebungsbedingungen anpassender Protokolle. Wir entwickeln daher Modelle zur Abbildung von Umgebungseinflüssen. Diese unterstützen die Konfiguration von Protokollparametern, so dass eine geforderte Zuverlässigkeit auch unter extremen Umweltbedingungen erreicht werden kann. Insbesondere charakterisieren wir die temperaturabhängige Abschwächung der Signalstärke und leiten ein plattformunabhängiges analytisches Model ab, das bei der Entwicklung temperaturkompensierender Protokolle zum Einsatz kommt. Darüber hinaus modellieren wir die statistische Verteilung von Funkstörungen zur Laufzeit und nutzen diese für den Entwurf von Protokollen.

Aufbauend auf den entwickelten Umweltmodellen, führen wir eine automatische Anpassung des Clear-Channel-Assessment-Schwellenwerts ein, um den Einfluss von Temperaturschwankungen auf Carrier-Sense-Multiple-Access-Protokolle zu verringern. Weiterhin erweitern wir bestehende MAC-Protokolle mit Mechanismen zur Verbesserung der Zuverlässigkeit gegenüber Funkstörungen und entwickeln das JAG-Protokoll, das als letzten Schritt der Handshake-Prozedur ein Störsignal einstellbarer Länge verwendet, um sicherzustellen, dass zwei benachbarte Knoten sich auch bei einer gestörten Funkverbindung zuverlässig auf ein gemeinsames Ergebnis einigen.

Riassunto

Il rapido sviluppo delle reti di sensori wireless – un insieme di dispositivi elettronici miniaturizzati a basso costo che consentono di raccogliere, elaborare e condividere informazioni dall'ambiente circostante – ha rivoluzionato, negli ultimi anni, il monitoraggio di fenomeni ambientali in molteplici ambiti applicativi. Come conseguenza del loro crescente impiego negli ambiti civile, scientifico, militare, agricolo e industriale, le reti di sensori wireless si accingono ora a pervadere la nostra vita quotidiana e a diventare parte integrante dell'Internet delle cose. Quest'ultimo consentirà, negli anni a venire, di accedere da remoto in tempo reale allo stato di qualsiasi "cosa" connessa in rete, indipendentemente dalla sua locazione fisica.

Dato l'elevato numero di potenziali utenti pronti ad utilizzare una miriade di attraenti servizi e di nuove applicazioni, l'aspettativa è che le reti di sensori wireless diventino sempre più affidabili. Se da un lato i valori rilevati devono essere recapitati in maniera affidabile e tempestiva, dall'altro una rete deve essere in grado di massimizzare la durata delle batterie dei vari nodi sensore ed operare autonomamente per diversi anni. Il mancato rispetto di questi requisiti potrebbe comportare scarsa soddisfazione degli utenti, danni economici e materiali, o perfino la perdita di vite umane.

Tuttavia, l'ambiente circostante ai nodi sensore gioca spesso un ruolo critico per le prestazioni della rete, in particolar modo per quanto riguarda l'efficienza e l'affidabilità delle comunicazioni radio fra i nodi sensore. Variazioni minime di temperatura possono infatti causare la perdita di sincronizzazione tra questi ultimi e causare un rapido degrado della qualità del segnale radio. Inoltre, interferenze provenienti da dispositivi radio limitrofi o altri apparecchi elettrici possono compromettere la corretta ricezione dei messaggi, riducendo la capacità di trasmissione effettiva e causando notevoli ritardi nello scambio di informazioni. Questa vulnerabilità all'ambiente circostante influisce sulla funzionalità di questi sistemi e non può quindi essere trascurata: le reti di sensori wireless devono soddisfare dei requisiti minimi di affidabilità per poter essere integrate nell'Internet delle cose.

La nostra tesi è che, attraverso una più profonda comprensione di come l'ambiente limitrofo influenza le prestazioni di una rete, si possano progettare protocolli di comunicazione più robusti ed efficienti, aumentando l'affidabilità delle reti di sensori wireless. In particolare, concentriamo la nostra attenzione sul ruolo delle variazioni di temperatura e delle interferenze radio e sviluppiamo in primo luogo *TempLab* e *JamLab*, strutture sperimentali a basso costo che consentono di riprodurre le problematiche introdotte dalle variazioni ambientali sull'efficienza e sulle prestazioni di una rete di sensori wireless.

Un secondo contributo di questa tesi è lo sviluppo di modelli computazionalmente leggeri in grado di catturare le caratteristiche dell'ambiente circostante con una pre-

cisione sufficientemente elevata da consentire di parametrizzare protocolli di comunicazione all'avanguardia e soddisfare determinati requisiti di affidabilità anche in presenza di condizioni ambientali difficili. In particolare, in questa dissertazione caratterizziamo l'attenuazione della potenza del segnale radio in funzione della temperatura e deriviamo un modello analitico indipendente dalla piattaforma utilizzata per progettare protocolli di comunicazione resistenti a continue oscillazioni termiche. Catturiamo inoltre la distribuzione statistica delle interferenze radio nell'ambiente circostante e deriviamo la probabilità che due nodi concordino positivamente l'esito di un trasferimento dati in canali congestionati.

Utilizziamo inoltre *TempLab* e *JamLab* per individuare ed evidenziare le limitazioni dei protocolli di comunicazione esistenti in presenza di condizioni ambientali variabili. I nostri risultati sperimentali evidenziano che la maggior parte dei protocolli di comunicazione è fortemente influenzata dalle variazioni di temperatura comunemente riscontrate in installazioni all'aperto, e che la trasmissione di dati a pacchetto non è adatta per compiere un *handshake* in canali radio congestionati.

L'obiettivo finale di questa tesi è la progettazione e lo sviluppo di protocolli che aumentino concretamente l'affidabilità delle reti di sensori wireless anche in presenza di condizioni ambientali ostili. In quest'ottica dimostriamo che, adattandone dinamicamente la *clear channel assessment threshold* alle variazioni termiche, possiamo migliorare significativamente l'efficienza dei protocolli ad accesso multiplo con rilevamento della portante in installazioni all'aperto. Estendiamo inoltre un protocollo MAC dimostrandone la maggiore robustezza ad interferenze radio e sviluppiamo *JAG*, un protocollo basato su una sequenza di *jamming* in grado di garantire che due nodi sensore adiacenti possano concordare sull'esito di un trasferimento dati nonostante la congestione del canale radio.

Contents

List of Figures

List of Abbreviations

AC Alternating Current

ACK Acknowledgement

ADC Analog-to-Digital Converter

AFH Adaptive Frequency Hopping

AGC Automatic Gain Control

ARQ Automatic Repeat reQuest

ATEX ATmosphères ed EXplosibles

CCA Clear Channel Assessment

CDF Cumulative Distribution Function

CRC Cyclic Redundancy Check

CSMA Carrier Sense Multiple Access

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

CSS Chirp Spread Spectrum

CTS Clear To Send

DAG Directed Acyclic Graph

DODAG Destination-Oriented DAG

DIFS Distributed Inter-Frame Space

DIO DODAG Information Object

EMI Electromagnetic Interference

ETX Expected Number of Transmissions

FEC Forward Error Correction

FHSS Frequency Hopping Spread Spectrum

GPRS General Packet Radio Service

GSM Global System for Mobile Communications

IEEE Institute of Electrical and Electronics Engineers

IFS Inter-Frame Spaces

IPv6 Internet Protocol Version 6

IR Infra-Red

ISM Industrial, Scientific and Medical

IoT Internet of Things

LPL Low-Power Listening

LPP Low-Power Probing

RPL IPv6 Routing Protocol for Low-Power and Lossy Networks

LQI Link Quality Indicator

MAC Medium Access Control

MIT Massachusetts Institute of Technology

NACK Negative Acknowledgement

PDU Protocol Data Unit

PI Proportional-Integral

PRR Packet Reception Rate

QoS Quality of Service

RAM Random Access Memory

RF Radio Frequency

RH Relative Humidity

RS Reed-Solomon

RSSI Received Signal Strength Indicator

RTS Ready To Send

SDR Software-defined Radio

SFD Start Frame Delimiter

SIFS Short Inter-Frame Space

SNR Signal-to-Noise Ratio

SPI Serial Peripheral Interface

TCP Transmission Control Protocol

TDMA Time-Division Multiple Access

UART Universal Asynchronous Receiver-Transmitter

UDP User Datagram Protocol

UWB Ultra-Wide Band

USB Universal Serial Bus

VGA Variable Gain Amplifier

VSG Vector Signal Generator

WSN Wireless Sensor Network

Chapter 1

Introduction

Since the first small-scale research projects emerged in the early years of the 21st century [139, 163], wireless sensor networks (WSN) have shown the potential to revolutionize our everyday life and to radically change modern society thanks to their ability to collect data at unprecedented spatial and temporal scales. Indeed, back in 2003, the MIT Technology Review periodical was presenting wireless sensor networks as one of the ten emerging technologies that will change the world, predicting their pervasiveness in every car, home, farm, building, office, and street [175].

Driven by this vision, the WSN research community came up with a large number of technical solutions: from the development of robust mote hardware [101, 165] and the creation of operating systems and programming languages tailored for this class of resource-constrained embedded devices [72, 85, 129], to the design of energy-efficient techniques for, among others, medium access control [75, 162, 164], data collection and dissemination [88, 130], localization [179, 197, 238], time synchronization [77, 81, 176], in-network aggregation [125, 137], and clustering [83].

The development of even more efficient protocols made wireless sensor networks an increasingly mature technology and enabled, in recent years, the long-term deployment of complete systems made up of hundreds of tiny battery-powered wireless nodes [150]. WSN technology has been successfully used to monitor, among others, civil infrastructures [55, 56, 119], urban environments [146, 180, 184], natural phenomena [30, 141, 222], agricultural and industrial processes [51, 121, 123], as well as unattended patients in clinical settings [37, 60, 122].

Several of these application domains impose strict *dependability* requirements on WSN performance. On the one hand, sensor data and actuation commands need to be reliably and timely delivered throughout the network. Sensor networks used to measure the vital functions of patients in hospitals [60] and to detect wildfire in forests and trigger alarms [15, 69, 95] are examples of safety-critical systems whose unreliability may result in injury or damage to the environment or even lead to loss of human lives. On the other hand, the energy consumption of wireless sensor nodes needs to be minimized, in order to maximize the lifetime of the system and to avoid a frequent battery replacement, which may result in an insufficient user satisfaction and high costs.

The aforementioned WSN installations and other research deployments have shown that the *environment* in which sensor nodes are embedded plays a critical role for their performance, especially with respect to their energy efficiency and the reliability of their low-power wireless communications. Radio interference from co-located wireless networks often affects the quality of links in most indoor installations, whereas varying meteorological conditions frequently reduce network performance in outdoor deployments.

The vulnerability of WSN systems to the surrounding environment should not be neglected, as most networks deployed nowadays are not small-scale research projects anymore, but rather civil and industrial deployments on a large scale. Furthermore, the deployment of a wireless sensor network is increasingly accompanied with its connection to the Internet. Cisco's Internet Business Solutions Group has indeed predicted that the number of devices that will be connected to the Internet will rise to 50 billion by 2020 [79], with the majority being small embedded devices with sensing capabilities.

This will result in an Internet of Things (IoT) in which low-power wireless sensor networks will represent the bridge between the physical and the digital world and will actually become an integral part of the daily life of millions of people. The IoT will therefore embrace a system of wireless networks that can deliver to end-users a plethora of services and attractive applications (e.g., smart cities, smart grids, and smart healthcare). At the same time, these systems will heavily rely on the dependable and predictable operation of networked embedded wireless sensors and actuators. Hence, in the years to come, wireless sensor networks will be expected to meet application-specific *dependability* requirements, and to minimize the impact of the environment on their performance.

1.1 Dependable Wireless Sensor Networks

Dependability is typically defined as the ability of a system to deliver the expected functionalities during its operational lifetime. Avizienis et al. [17] have broken down the notion of dependability into three elements: attributes, threats and means. *Attributes* are qualities of a system that can be used to determine its overall dependability: a dependable system is typically expected to be operational during its envisioned lifetime (availability), to operate correctly (reliability), to cause no unauthorized disclosure (confidentiality) or modification of information (integrity), to have the ability to undergo modifications and repairs (maintainability), as well as to operate in a harmless fashion (safety).

Threats are issues that can affect the correct operation of a system. Faults are typically defects in a system, but their presence might not necessarily lead to a drop in dependability. Only once a fault is activated an error occurs, resulting in a discrepancy between the expected and the actual system behaviour. Unless properly handled, errors can lead to failures, i.e., to situations in which a system or component is unable to perform its required functions within specified performance requirements.

To attain the various dependability attributes, several *means*, i.e., mechanisms intended to reduce the number of failures presented to the user of a system, can be developed. Faults can be prevented from being incorporated into the system (fault prevention), predicted before they occur (fault forecasting), or removed at runtime (fault removal). Another possibility is to develop mechanisms allowing a system to still deliver the required service in the presence of faults, possibly at a degraded level (fault tolerance).

This doctoral thesis is devoted to the design of solutions that increase the dependability of wireless sensor networks deployed in harsh environments. With respect to the *environmental threat* affecting WSN performance, this work specifically focuses on two dependability attributes: the *reliability* of low-power wireless communications, and the energy-efficiency of networking protocols, as the latter strongly affects the lifetime of the system and hence its *availability*.

Reliability is commonly defined as the probability that, under stated conditions, a system will correctly perform its intended function(s) during a specified period of time. In wireless sensor networks, the primary task of each sensor node is to successfully deliver sensor data and actuation commands throughout the network. Doing that reliably and timely is of utmost importance, given that wireless sensor networks are often used to perform critical tasks such as earthquake prediction, medical monitoring, object tracking, civil infrastructure monitoring, and car-to-car communication. To build dependable wireless sensor networks, communication protocols should hence be able to sustain a high delivery rate despite external influences (i.e., they should be *robust* to external factors).

The availability of a system is commonly defined as the ratio of the up-time U and the aggregate of up- and down-time $(U + D) = L$, with L the expected (or desired) system lifetime. A system is hence highly available if U/L tends to one, i.e., if its down-time D is minimal. In the context of wireless sensor networks, an important factor affecting D is the speed at which the batteries of sensor nodes are depleted. To build dependable WSN, communication protocols should hence be highly *energy-efficient* in order to maximize the duration of batteries and allow the network to run for the envisioned amount of time.

1.2 Problem Statement

In this doctoral thesis we aim to increase the dependability of wireless sensor networks deployed in harsh environments by means of reliable and available (i.e., energy-efficient) communication protocols. To achieve this goal, a first key challenge is to obtain an accurate understanding of *how* the surrounding environment affects WSN hardware and protocols.

Characterizing the environmental impact on WSN operation. Real-world deployments have reported different network performance across the 24-hours that correlate, among others, with the presence of people and their activities (indoors) and with the varying meteorological conditions (outdoors).

Outdoor environments. Wireless sensor networks deployed outdoors are often affected by time-varying environmental conditions, such as the presence and density of vegetation [86, 140], and variations in temperature [23, 33, 220] or humidity [143, 207]. Different types or a diverse distribution of foliage across different seasons can change the physical environment surrounding the nodes, leading to fading and shadowing that can substantially change the connectivity in the network. Meteorological conditions can affect radio propagation: rainfall and snow can create pools of water or thick snow covers that may cause up to 30 dB fades [52, 196], whereas nodes that are not fastened properly may not withstand wind blasts and suffer from vibrations that may reduce sensing accuracy [32, 230].
Furthermore, diurnal and seasonal temperature variations can drastically reduce network performance, as they can cause loss of synchronization [182], degradation of the wireless link quality [23], and early battery discharge [159]. Temperature variations in outdoor

installations are particularly relevant since they have a strong impact on the operations of all electrical and electronic components (whereas the impact of vegetation and meteorological conditions is highly specific to the setup and location of the deployment and may require individual studies). Problems especially occur when wireless sensor nodes are exposed to direct sunlight or are enclosed into transparent packaging absorbing IR radiation, as the on-board temperature of each node can reach values that are significantly higher than air temperature (up to 70 °C according to real-world observations [29, 33, 34, 166]). Balendonck et al., for example, have experienced a complete failure of their irrigation management system when the repeater nodes were exposed to direct sunlight [21]. In addition to high temperatures, also a high variability across different nodes in the network can have a negative impact on network performance. High gradients of temperature caused by the presence of vegetation and by clouds or obstacles blocking the sun radiation can be as high as 25 °C for nodes that are spatially close to each other [45, 208, 209], and may lead to de-synchronization among nodes. Hasler et al. have indeed reported problems in maintaining time synchronization across their network deployed on the Swiss Alps in the presence of large temperature fluctuations [96].

Indoor environments. Wireless sensor networks deployed in indoor environments can have a highly-variable wireless link quality [20]. Temporal variations in the quality of wireless links often result from changes in environmental conditions that exacerbate fading and shadowing effects, such as the ones caused by metal doors and other obstacles between sensor nodes, or by the presence of people moving across the building [135].
The most serious threat for wireless sensor networks deployed indoors is, however, radio interference. Wireless sensor networks use the industrial, scientific and medical (ISM) radio bands for their communications – freely-available unregulated portions of the radio spectrum. An increasing number of devices share these unlicensed frequencies, and several studies have shown that the interference from wireless devices and other electrical appliances located in the surroundings of sensor nodes often impairs their low-power communications, reducing throughput and leading to an increased amount of network traffic due to retransmissions [161, 189, 240]. In addition to a decrease in throughput, radio interference can also lead to high latencies, to a complete lack of connectivity among some of the nodes in the network, as well as to an increase in energy consumption due to retransmissions and longer wake-up times that may cause an early battery depletion [38]. To date, the 2.4 GHz ISM band is by far the most congested, as the communications of wireless sensor networks and other IEEE 802.15.4 devices operating in these frequencies have to coexist with the transmissions of Wi-Fi (IEEE 802.11) and Bluetooth (IEEE 802.15.1) devices, as well as with the radio frequency (RF) noise generated by microwave ovens and other domestic appliances such as cordless phones, baby monitors, game controllers, presenters, and video-capture devices [42, 240]. Reports from real-world deployments have highlighted that the congestion can be quite significant in residential and office buildings [185, 186], and that, in the vast majority of the cases, the interference sources cannot be easily identified [24].

Considering that radio interference and temperature variations have a profound impact on the dependability of indoor- and outdoor-deployed wireless sensor networks, this thesis focuses primarily on these two environmental factors, and aims to precisely characterize their impact on WSN hardware and communication protocols.

Lack of testbed infrastructures with realistic environment effects. Characterizing the environmental impact on wireless sensor networks can be quite challenging, as there is a lack of proper testbed infrastructures that can be used to reproduce specific environmental conditions. On the one hand, simulation tools cannot capture the complexity of the real world, and analytical models that can improve their efficiency can only be derived based on a precise understanding of the environmental impact on sensornet hardware, which is hard to obtain due to the complexity of the involved physical processes. On the other hand, testing in real-world deployments is a costly, time-consuming, and labour-intensive operation that often does not shed light on the poor performance of a wireless sensor network. Indeed, the impact of temperature cannot be isolated properly when testing WSN operations at the deployment site: meteorological conditions cannot be controlled, and the temperature profiles that can be tested are highly specific to the deployment location, to the position of the nodes, and to the time of the year in which the experiment is carried out. Similarly, the impact of interference is highly stochastic in nature, and strongly depends on the physical environment surrounding the nodes (e.g., on the position, traffic pattern, and manufacturer of the interfering devices), which may not be controllable.

Hence, one goal of this Thesis is the development of low-cost experimental facilities that do allow researchers and system designers to replay the impact of the environment (and specifically of radio interference and temperature variations) in a fast and simple way, and that do offer the possibility to accurately repeat the same experiment in order to thoroughly characterize the performance of communication protocols and obtain sound performance comparisons.

Dependable protocols despite adverse environments. As the employment of wireless sensor networks is moving from mere monitoring applications to critical domains such as traffic control [108], smart cities [180], smart health [37], and the smart grid [50], achieving dependable operations despite the presence of radio interference and temperature variations becomes of utmost importance. Indeed, developers cannot create smart city solutions if parking spots occupancy and pollution concentration sensors are not operating as expected during the hottest times of the day or in the presence of radio interference in dense urban environments. Patients wearing body sensor networks measuring their vital functions cannot rely on alarms being promptly posted to the medical staff if the sensor nodes behave differently in indoor and outdoor environments, or if a nearby Wi-Fi access point blocks their communications while they are at home or in a lively street. Similarly, wireless sensor systems used to detect wildfire in forests and rapidly trigger alarms cannot afford to fail when temperature suddenly increases as a consequence of a fire front spreading in the proximity of the nodes. A wireless sensor network should instead operate reliably and efficiently regardless of the climate or of the number of wireless devices operating in the surroundings. Failing to do so may lead to malfunctioning protocols and to applications or products that are not well-received by final users or customers or that result in injury and damage to the environment.

The ultimate goal of this thesis is hence to design protocols that can mitigate the impact of radio interference and temperature variations on wireless sensor networks, which is a fundamental step required to bring the visions and promises of the future Internet to reality.

1.3 Contributions

The scientific contributions of this thesis in the area of low-power wireless networking are summarized in Table 1.1. In addition, the work presented in this dissertation has produced software that is publicly available [2] and has been largely used by the research community.

Testbeds with realistic environmental effects. The first contribution of this thesis is the design and implementation of low-cost testbed infrastructures that enable the repeatable playback of specific environmental conditions, namely temperature and radio interference. In particular, TempLab [45] and JamLab [42] are extensions for WSN testbeds that allow, respectively, to easily study the impact of temperature on sensornet hardware and protocols, and to create realistic and repeatable interference patterns. These low-cost extensions allow to rerun experiments under almost identical environmental conditions and hence play a crucial role in the investigation of protocol performance.
TempLab can accurately reproduce temperature traces recorded in outdoor environments with an average error of only 0.1°C, and can be used with specific test patterns (e.g., a series of cold and warm periods) to allow a quick debugging of protocol behaviour. TempLab can also reproduce model-based temperature profiles to have an estimation of the temperature dynamics at a certain location without the need of traces collected in-situ.
JamLab is a low-cost infrastructure that augments existing sensornet testbeds with accurate interference generation by using off-the-shelf sensor nodes. It provides lightweight interference models of devices commonly operating in the 2.4 GHz ISM band, as well as a playback capability to regenerate previously recorded interference patterns.

Models characterizing the environmental impact. The second contribution of this thesis is to study and model *how* the environment affects WSN operations.
We use TempLab to observe and quantify the impact of on-board temperature variations on sensornet performance. Controlled experiments backed-up with outdoor measurements show that off-the-shelf low-power wireless transceivers can experience a significant decrease in received signal strength at high temperatures [34]. This insight is particularly relevant since the variations of the on-board temperature of wireless sensor nodes deployed outdoors can be quite high, especially if nodes are enclosed in industrial packaging and IR-transparent enclosures [33, 34]. We exploit TempLab also to characterize the attenuation in signal strength as a function of temperature on several low-power wireless transceivers experimentally, and derive a platform-independent analytical model that is later used to design temperature-aware protocols [44].
Radio interference is modelled using a simple two-state Markov model. This model gives sensor nodes the ability to record and replay the interference patterns generated by common wireless devices operating in the 2.4 GHz ISM band – a fundamental pillar for the design of JamLab [42]. Furthermore, this lightweight interference model enables us to capture the statistical distribution of interference with sufficiently high accuracy even on resource-constrained sensor nodes, as it requires only minimal storage requirements. We then show that knowledge about the statistical distribution of interference allows us to parametrize protocols parameters such that certain QoS requirements are met even in the presence of external interference [46].

Challenge	**Contribution**	
	Temperature	***Radio interference***
Lack of testbed infrastructures	• TempLab (Section 3.1)	• JamLab (Section 4.2)
Characterization of environmental impact	• Impact on packet reception (Section 3.2.1) • Impact on signal strength attenuation and platform-independent model (Section 3.2.2 and 3.2.3) • Impact on communication protocols (Section 3.3)	• Characterization of common interference sources (Section 4.1.2) • Lightweight interference models (Section 4.1.3) • Performance of state-of-the-art MAC protocols (Section 4.3.1)
Dependable protocols despite adverse environments	• Temperature-aware MAC protocols (Section 3.4)	• Enhanced X-MAC (Section 4.3.2) • JAG – Jamming-based AGreement (Section 4.4)

Table 1.1: Overview of the contributions of this thesis.

Shortcomings of existing WSN protocols. The third contribution of this thesis is to identify and highlight several limitations of existing communication protocols in the presence of varying environmental conditions.

TempLab is used to show that state-of-the-art communication protocols can be strongly affected by on-board temperature variations commonly found in outdoor deployments, as they neglect the attenuation that temperature has on the received signal strength in low-power wireless radios. Experimental results indicate that (i) routing protocols can experience drastic changes in the topology of the network, including some temporary partitions and large increases in diameter [45], and that (ii) data link layer protocols may experience a reduced effectiveness of clear channel assessment at high temperatures that compromises the ability of a node to avoid collisions and to successfully wake-up from low-power mode [41].

Taking advantage of JamLab, we compare the performance of several state-of-the-art MAC protocols under interference [43]. We show that specific protocol features such as rendezvous schemes preceding the actual data transmission and the choice of congestion back-off schemes play a critical role in these settings and may significantly affect packet reception and energy-efficiency. Furthermore, this thesis focuses on the agreement problem in congested environments, i.e., on how to agree on fundamental pieces of information in environments with high packet loss rate. Our experimental analysis shows that traditional packet-based handshakes are not suitable in environments rich of radio interference, as they often lead to a large fraction of disagreements and excessively large energy expenditures [46].

Environment-aware communication protocols. Further contribution of this thesis is the design and implementation of environment-aware protocols that increase the dependability of wireless sensor networks deployed in harsh environments.
To mitigate the impact that temperature variations have on carrier sense multiple access protocols, two mechanisms that dynamically adapt the clear channel assessment threshold to temperature changes are developed, thus making data link layer protocols temperature-aware [41]. An extensive experimental evaluation carried out using TempLab shows that both approaches considerably increase the dependability of a network in the presence of temperature variations commonly found in outdoor deployments, with up to 71% lower energy consumption and 194% higher packet reception rate [41].
To mitigate the effects of radio interference on low-power wireless communications, we augment an existing duty-cycled MAC protocol with mechanisms that improve its robustness to interference while remaining reasonably energy efficient [43]. Furthermore, we develop JAG [46], a protocol that uses a jamming sequence of configurable size as a last iteration of an handshake to make sure that two neighbouring nodes agree on a given piece of information. A thorough experimental analysis shows that JAG not only outperforms message-based approaches in terms of agreement probability, energy consumption, and time-to-completion, but that it can also be used to obtain performance guarantees and meet the requirements of applications with real-time constraints.

1.4 Scientific Impact

The contributions of this thesis have been published in top-tier conferences, journals, and books, most notably in the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), the IEEE International Real-Time Systems Symposium (RTSS), the European Conference on Wireless Sensor Networks (EWSN), the IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), and the IEEE Transactions on Industrial Informatics.
The paper "Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers" published at the 5^{th} Extreme Conference on Communication (ExtremeCom) received the Best Paper Award.
The paper "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation" published at the 10^{th} International Conference on Information Processing in Sensor Networks (IPSN) was a Best Paper Runner-Up.
The demo "How Temperature Affects IoT Communication", highlighting the results presented in two of the papers included in this thesis, was a Best Demo Runner-Up at the 11^{th} European Conference on Wireless Sensor Networks (EWSN).

Lastly, the JamLab tool has been used by a number of international researchers working in the WSN field to evaluate their protocols, and their contributions have been published in several top-tier scientific conferences. These contributions include a burst forwarding technique to allow high throughput data transport in lossy wireless networks [74], an extension of low-power listening that performs channel hopping [147], a contention resolution mechanism for low-power wireless networks [155], as well as a technique to select the best access point available to transfer data in the presence of interference [82]. Furthermore, components of JamLab have been used to measure the characteristics of interference and design channel quality metrics [149] and to estimate the packet reception rate in noisy environments and select an optimal packet size [47]. Similarly, TempLab is currently being used by a number of international researchers to analyse the limitations of state-of-the-art routing protocol in the presence of temperature variations and evaluate the efficiency of the proposed solutions [117, 242].

1.5 Methodology

This doctoral thesis follows an experimentally-driven methodology. We first carry out a thorough literature review, and analyse the impact of temperature variations and radio interference on the reliability of low-power wireless communications and on the energy-efficiency of networking protocols. We then increase the dependability of wireless sensor networks by *means* of forecasting and tolerance.

The goal of forecasting is to predict faults, and to remove them before they occur or to circumvent their effects. To predict and correct faults, we require accurate models capturing the environmental impact on WSN performance. We therefore characterize the performance of wireless sensor networks as a function of the environment, and enhance existing protocols or design novel solutions that can neutralize its impact. In many situations, however, the environmental impact cannot be properly anticipated or corrected. In this case, we increase the dependability by means of tolerance, i.e., we make sure that a system can still operate correctly also in the presence of faults (if necessary, at the price of a decrease in performance).

To achieve this goal, a fundamental stepping stone is the creation of testbed infrastructures enabling the repeatable playback of specific environmental conditions, namely temperature and radio interference. These testbeds systematically “replay” the same conditions found in the real-world, allowing us to derive general models capturing the impact of the environment on WSN performance and to analyse and test the behaviour of communication protocols, as shown in Figure 1.1. While designing and building these testbed facilities, special care is put on two aspects. The first one is the precise isolation of a specific environmental factor in order to avoid any sources of bias. TempLab, for example, isolates the effects of temperature on WSN hardware, in contrast to most outdoor deployments and testbeds, where sensor nodes are not only exposed to temperature variations, but also to a number of meteorological conditions that may accentuate or contribute substantially to the performance degradation. The second aspect is the ability to replay similar conditions to the ones actually occurring in the real-world.

Using these testbed facilities, we derive models that accurately capture the impact of radio interference and temperature on WSNs. While devising these models, we pay special attention to two aspects. First, models need to be computationally lightweight, so that they can be exploited by resource-constrained wireless sensor nodes. Second, they should allow to design solutions that can pro-actively neutralize or mitigate the environmental impact on WSN performance. Please notice that one can also increase the dependability of

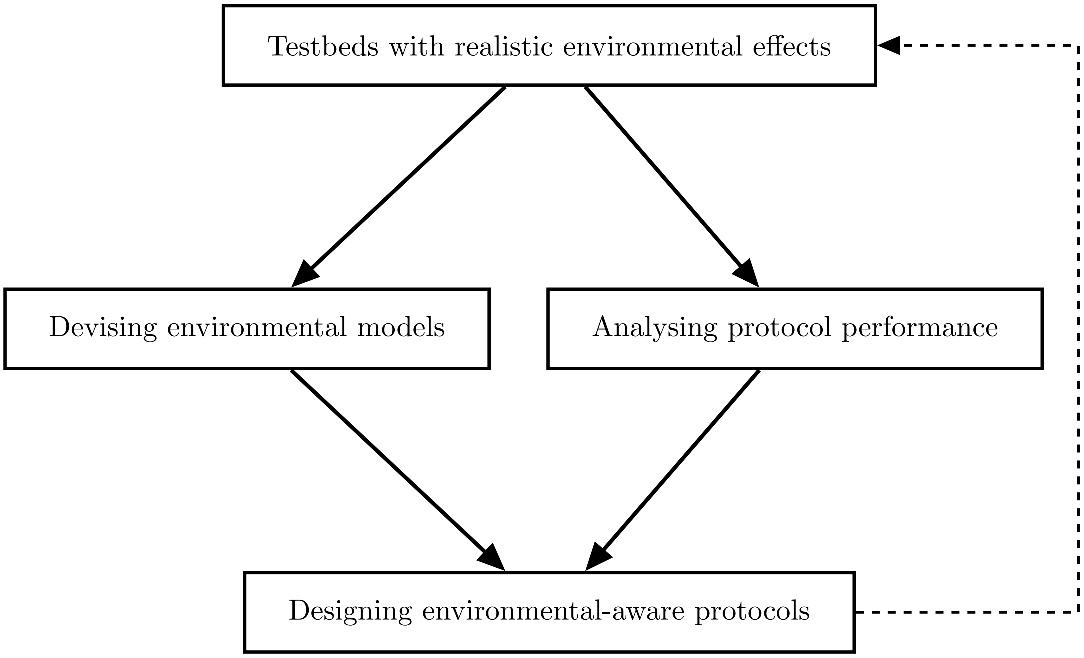

Testbeds with realistic environmental effects
Devising environmental models
Analysing protocol performance
Designing environmental-aware protocols

Figure 1.1: Experimentally-driven methodology followed in this thesis.

a system without resorting to an environmental model. However, model-free solutions do not offer the ability to pro-actively prevent performance deterioration, and reacting to a loss of performance may not fit the requirements of most safety-critical WSN applications.

We further employ the testbed facilities to analyse the performance of existing communication protocols. First, we formulate a hypothesis about the (erroneous) behaviour of the examined protocol in the presence of a given environmental parameter. We then select ad-hoc or suitable real-world traces (of temperature or radio interference) that can exacerbate the problem, and replay them in the testbed. Using several software tools, most notably the Contiki operating system [72], the impact on state-of-the-art communication protocols is systematically analysed. If a decrease in performance is observed, the protocol is thoroughly studied and an increasing amount of debug information collected, until the source of the performance reduction or the design flaw is identified. If the experiments, instead, falsify the hypothesis, we either change the latter accordingly and refine the experiments, or the idea is dropped altogether.

Building upon the devised environmental models and upon the lessons learnt while assessing the performance of existing protocols, we either augment existing protocols or conceive a completely new design. The performance of the augmented or newly-designed protocols is then carefully tested under different environmental conditions to make sure that (i) their behaviour matches the expectations, and that (ii) they actually increase the reliability and availability of WSN deployed in harsh environments. To prove the efficacy of the implemented solutions, the ability of the testbeds to "replay" the same experiment multiple times has been fundamental, and also led to relevant insights and triggered new research ideas.

1.6 Structure

The rest of this dissertation is organized as follows. Chapter 2 discusses existing work in the area and highlights the challenges in characterizing and mitigating the impact of the environment on wireless sensor networks, with specific focus on temperature and radio interference. These two environmental factors are then addressed separately in chapters 3 and 4, respectively. First, the design of testbed infrastructures with realistic environmental effects is described, with special emphasis on how these are used to characterize the environmental impact on WSN hardware and protocols. Building upon these insights, the design and implementation of environment-aware protocols is then presented, along with a thorough evaluation of their performance in extreme conditions. Chapter 5 concludes this doctoral thesis by summarizing the obtained results beyond the state of the art, by discussing limitations, and by providing an outlook on future work.

Chapter 2

Related Work and Research Challenges

This chapter reviews the body of work addressing the impact of the environment on WSN performance. Section 2.1 summarizes real-world experiences highlighting deployment failures and problems triggered by hostile environmental conditions. The following two sections detail the impact that temperature variations (Section 2.2) and radio interference (Section 2.3) have on wireless sensor networks, and examine the solutions that were proposed by the research community to increase their performance despite these adverse conditions. The aim is not to be exhaustive, but rather to accurately categorize existing solutions, and to analyse the yet open challenges. Finally, Section 2.4 describes experimentation in WSN testbeds with emphasis on the repeatable playback of specific environmental conditions, and points out the lack of proper infrastructures when it comes to experimenting with temperature and radio interference.

2.1 Environmental Impact on WSNs

Embedding wireless networks of sensor nodes directly into the environment allows to autonomously monitor real-world phenomena at an unprecedented scale. This has triggered, in the last two decades, the installation of WSN systems in residential and office buildings [8, 66], hospitals [37, 60, 122], urban environments [146, 180, 184], civil infrastructures [55, 56, 119], agricultural fields and vineyards [12, 26, 51, 121, 126], forests [54, 69, 209], industrial settings [123], as well as harsh and hardly-accessible areas such as volcanoes [222, 223], deserts [23], remote islands [139], glaciers [141], high mountains [25, 30], and polar research facilities above the Arctic circle [221].

Several of the aforementioned installations and other research deployments have shown that the environment in which sensor nodes are embedded plays a critical role for their performance, and triggered a number of research efforts to increase the dependability of WSN systems [3, 4, 6]. A large number of researchers have indeed observed different network performance across the 24 hours that correlate very often with the presence of people and their activities (indoors) as well as with the varying meteorological conditions (outdoors).

In most indoor installations, human-related activities strongly affect the quality of links and the reception of packets [46, 80, 135]. People moving in the surroundings of sensor nodes often exacerbate fading and shadowing effects, leading to fluctuating link qualities. Lin et al. [135] have identified small fluctuations caused by multi-path fading of wireless signals, large fluctuations caused by shadowing effect of humans, doors, and other obstacles, as well as continuous large fluctuations caused by the interference and noise generated by surrounding wireless devices and home appliances such as Wi-Fi access points and microwave ovens [20]. Especially the impact of radio interference can be quite severe, as it may intermittently increase the packet loss rate and the number of retransmissions during daytime in office environments and during evenings and weekends in residential buildings [240]. Indeed, any wireless device operating at higher power than sensor nodes in the same unlicensed ISM frequency bands may emit RF signals that can block the low-power communications of sensor nodes [192].

Networks deployed outdoors often experience a reduction in the packet reception rate and in the transmission range between sensor nodes in the presence of fog [11], rain [33], and thick snow covers [52, 196]. The performance of outdoor WSN systems is also correlated with variations in temperature [23, 34, 220] and relative humidity (RH) [143, 207]. Temperature variations can affect clock drift and hence time synchronization, leading to connectivity issues. Furthermore, they can also affect battery capacity and discharge, reducing the lifetime of a network. Some deployments have experienced a complete system failure at high temperatures [21], or reported drastically worse performance during the hotter times of the day [23, 32] or year [220]. A few outdoor deployments have also shown radically different performance across seasons and linked them to the variable density and distribution of foliage in the surroundings of sensor nodes [86, 140].

Environmental changes may also affect the reliability and accuracy of sensing in both indoor and outdoor environments. Nodes that are not fastened properly may not withstand wind blasts and suffer from vibrations affecting the meaningfulness of sensor readings [32, 230]. Similarly, the accuracy of sensor data measured on mobile nodes that are occasionally exposed to direct sunlight may be reduced, e.g., when people wear body sensors outdoors [36].

Furthermore, harsh environmental conditions can have a strong impact on the employed hardware and cause physical damage to the sensor nodes [24, 31, 172]. In their Génépi rock glacier deployment, Barrenetxea et al. [24] have experienced corrosion of a sensor connection that made the collected data unusable. Similarly, O'Donovan et al. [158] have shown that the industrial environment in which sensor nodes were deployed (the Petrogal oil refinery in Sines, Portugal) was highly corrosive and especially affected the external antennas. In their Great Duck Island deployment (a small island in the Gulf of Maine, USA), Polastre et al. [166] have reported an early battery depletion due to the low-resistance path between the power supply terminals created by water entering the packaging [31]. As a side effect, also erroneous sensor readings were observed. Similarly, Tateson et al. [203] have experienced water leakage in one of the deployed sensors, due to the last-minute software changes leading to a reconnection of the sensor cables, which caused a complete failure of the sensor modules. In GlacsWeb [141], the authors report outages of the base station and speculate that two of the possible reasons are the rupture of the casing due to the moving ice, or extremely large clock drifts hindering a reliable communication.

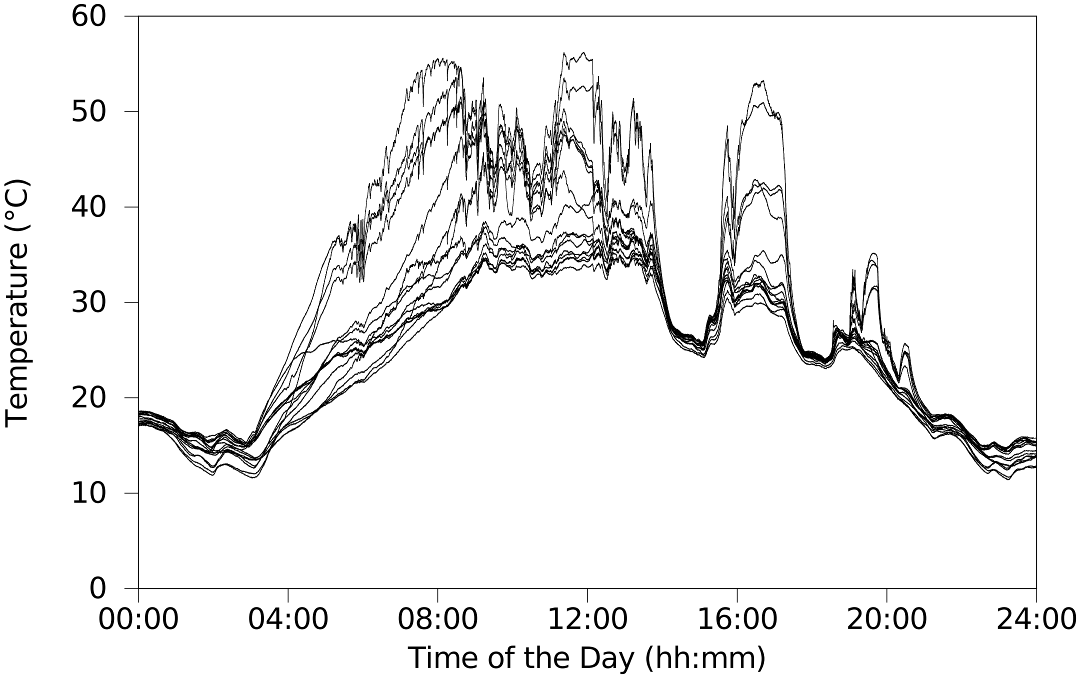


Figure 2.1: On-board temperatures recorded within 24 hours by sixteen TelosB nodes deployed in an outdoor setting during summer [45] The temperature difference across the nodes can be as high as 30 °C.

2.2 Impact of Temperature on WSN Performance

Depending on the packaging and on deployment location, the electronics of wireless sensor nodes may experience substantial temperature variations over time and space. The latter can have a detrimental effect on network performance, as they can significantly affect clock drift [24, 96, 182], battery capacity and discharge [159], as well as the efficiency of low-power radios [23, 34]. This section analyses in detail the typical on-board temperature fluctuations observed in research and industrial deployments (Section 2.2.1) and points out how they can lead to loss of synchronization (Section 2.2.2), early battery depletion (Section 2.2.3), and degradation of the wireless link quality (Section 2.2.4), which would in turn affect key dependability properties such as reliability and availability. Section 2.2.5 identifies a number of challenges that have not been yet addressed by the research community.

2.2.1 Temperature fluctuations in real-world deployments

The on-board temperature of sensor nodes is often higher than air temperature measured by traditional weather stations, and can drastically vary over time and space [166].

High temperatures. Deployments of wireless sensor networks at high altitudes have observed that on-board temperature variations across different times of the year can be

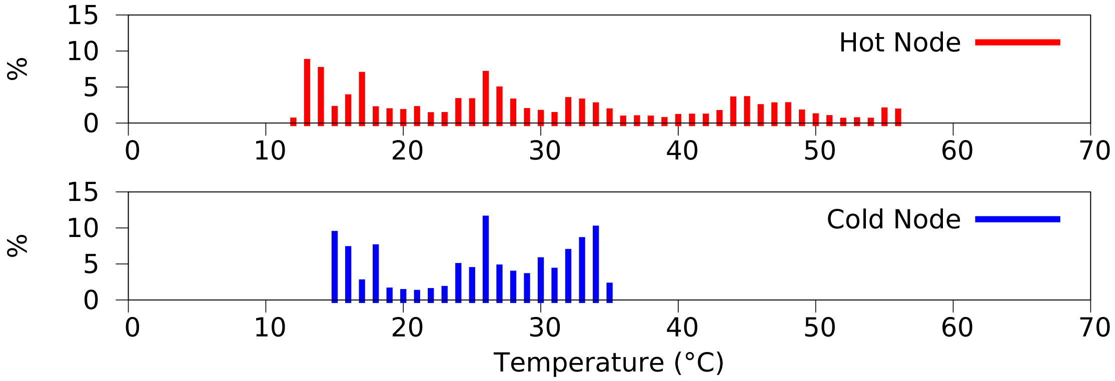


Figure 2.2: The on-board temperature profile of wireless sensor nodes deployed outdoors can be highly different even if nodes are in close proximity of each other [45].

extremely large: Beutel et al. have shown that temperatures may vary from -40 °C in wintertime to +60 °C during summer [29, 30]. Sunshine may indeed easily heat a packaged sensor node up to 70 °C – especially if the packaging absorbs infra-red (IR) radiation [33, 166, 199]. Airtight and waterproof enclosures may protect the node from corrosion, humidity and atmospheric contaminants [24, 31], but may significantly increase the temperature of the inner casing [33, 166]. Nodes experiencing high temperatures (e.g., exposed to direct sunlight or enclosed into IR-transparent enclosures) are more prone to failures: Balendonck et al. [21] have suffered a complete failure of their irrigation management system, whereas Beutel et al. [30, 141] have experienced reliability problems of electrical components such as oscillators and analog-to-digital converters (ADC).

Temperature gradients. Temperature introduces a dynamic heterogeneity across the network that can have a negative impact on network performance. On-board temperature gradients are typically caused by the presence of vegetation, clouds, buildings, or obstacles blocking the sun radiation, and a number of deployments have shown that nodes that are spatially close to each other may experience completely different temperatures [208, 209, 220]. Figure 2.1, based on traces recorded by Wennerström et al. during a long-term outdoor deployment in Uppsala, Sweden [220], shows the on-board temperature of sixteen TelosB nodes over the course of a summer day. The temperature difference across the nodes (which were placed within each other's transmission range at a maximum distance of 80 meters, and exchanged packets while periodically recording their on-board temperature) can be higher than 25°C, and some nodes are significantly "hotter" than others. Figure 2.2 illustrates the temperature density function for two nodes in this deployment: one node experiences temperatures between 12 and 56 °C, whereas the on-board temperature of the other node varies by at most 20 °C across the 24 hours.

2.2.2 Clock drift and time synchronization

The resonating frequency of a quartz crystal can substantially vary with changes in ambient temperature. To act as resonator, a piece of quartz is typically cut at specific angles with respect to the crystal grid. When subject to a proper alternating voltage, the crystal begins to vibrate and produces a sinusoidal signal. The frequency of this signal depends on the type of cut and is determined by the thickness, density, elasticity, and area of resonance over which the quartz plate is operating [100].

These factors are influenced, among others, by temperature changes: crystal oscillators commonly resonate close to their target frequency at room temperature, but any deviation in temperature will cause them to slow down or accelerate. A crystal oscillator with a target frequency at room temperature could experience frequency offsets as high as 25 *ppm* for a temperature variation of 5 °C, a non-negligible change given the high fluctuations occurring in outdoor environments [100]. A survey of packaged quartz crystal resonators has revealed that the majority of them experience a drift over the temperature range [-20, 50] °C of at least 50 *ppm*, a variation sufficient to affect the rendezvous process of synchronous duty-cycled MAC protocols [182].

In the presence of high fluctuations of the on-board temperature over time and across different nodes as shown in Section 2.2.1, clock drift becomes non-negligible, and several WSN systems have experienced malfunctioning. While measuring permafrost processes in the Swiss Alps, Beutel et al. have observed large spatio-temporal variations in temperature causing a clock drift that triggered several system resets [29, 96]. To allow a proper tuning of the timing-dependent operations executed by the processor, an extensive testing in thermal chambers was required, as well as the design of a clock-drift compensation mechanism. In another deployment at high altitudes [24, 25], the drift caused by rapid temperature changes resulted in a loss of synchronization between the sensor node and the GPRS modem, and hence in lost packets. The problem especially occurred during sunrise, when temperature gradients up to 10 °C per hour were observed.

To compensate for the clock drift, several researchers have proposed the use of a local temperature sensor to autonomously calibrate the local oscillator and remove the effects of temperature variations. Schmid et al. [183] have developed a temperature-compensated time synchronization technique that exploits local temperature readings to autonomously learn the calibration parameters of the local crystal and provide a stable temperature-compensated crystal oscillator. Similarly, Brunelli et al. [48] presented a low-overhead temperature compensation algorithm for clock synchronisation in wireless sensor networks, achieving an effective clock drift of less than 5 *ppm* over a wide range of operating temperatures.

Correcting the drift based on local temperature measurements is very popular in wireless sensor networks (comparable approaches were also proposed in [29, 53, 235]), as almost every sensor node carries a temperature sensor on-board. Even if a sensor node does not have a dedicated on-board temperature sensor, several low-power micro-controllers such as the MSP430 offer the possibility to obtain a rough estimate of the on-board temperature from a built-in temperature sensor using a specific input of the ADC and can hence be used to correct the drift.

2.2.3 Battery capacity and discharge

Batteries typically operate best at room temperature, and any change (hot or cold) affects their performance or longevity. Higher temperatures increase the mobility of the electrolyte materials, resulting in a lower internal resistance and a significant increase in the effective capacity of the battery. On the contrary, lower temperatures increase the internal resistance of the battery and hence reduce its capacity. Furthermore, continuous temperature fluctuations also exacerbate self-discharge effects, which cause an irreversible loss of charge when no current is drawn from the battery.

These effects are highly relevant, as the vast majority of WSNs are battery-powered and as the nodes in a network may experience completely different temperature profiles. In the example shown in Figure 2.2, the "colder" node would have a different lifetime compared to the "hotter" one [159], and being able to predict the impact of temperature variations on battery discharge would allow a better mitigation of the problem.

A number of works have characterized the thermal impact on battery discharge experimentally [91, 159]. Park et al. [159] have presented a detailed experimental study on the role played by temperature and protocol parameters on the efficiency with which wireless sensor nodes drain their batteries. Guo et al. [91] have experimentally derived the distribution of voltage levels at different temperatures on several batteries, quantifying the thermal effects on lifetime.

The latter is often used as a measure of remaining battery capacity, and Ceriotti et al. [55] have recorded the evolution of the battery levels in conjunction with the measured ambient temperature over a 7-month period in a road tunnel deployment. Their diagrams show that temperature changes affect the voltage of the battery: a series of warmer days in winter, for example, coincides with a slight recovery and increase of the voltage. Similarly, Sun and Cardell-Oliver [198] have highlighted fluctuations up to $0.25\,V$ in the battery voltage along the 24 hours due to the temperature variation between day and night. These fluctuations complicate the estimation of the remaining battery capacity, and an accurate battery modelling as a function of temperature is hence an active research topic [173].

2.2.4 Inefficiency of low-power radios

The impact of temperature on electric conductors and semiconductors is well-known. In electric conductors, a higher temperature increases the resistance of the medium, whereas in semiconductors, it leads to current leakage. In other words, for a given voltage, a higher temperature reduces the current and hence the power of a device.

A few studies have shown that the output and received power of common low-power radio transceivers is affected by temperature variations. Yamashita et al. [233] have shown the first evidence that temperature decreases the efficiency of RF circuits on their ultra-small, low-power sensor-net module called ZN1, equipped with the CC2420 transceiver. The authors have enclosed the ZN1 modules in a thermal chamber and monitored the change of RF characteristics and current consumption while the temperature was changing, observing a decrease in the RF output power by about 5 dB at 80 °C compared to an initial temperature of 25 °C.

Similarly, Bannister et al. [23] have quantified the decrease in efficiency caused by temperature changes on the CC2420 platform. In their experiments in a climate chamber, they observed a decrease of 4-5 dB in the output power of the transmitter and a drop of 3-4 dB in the received power over the temperature range 25-65 °C, for a combined effect on received signal strength of about 8 dB when both transmitter and receiver nodes are heated. The authors argue that this attenuation is the result of the decreased efficiency of the transmitter's power amplifier and the receiver's low-noise amplifier at high

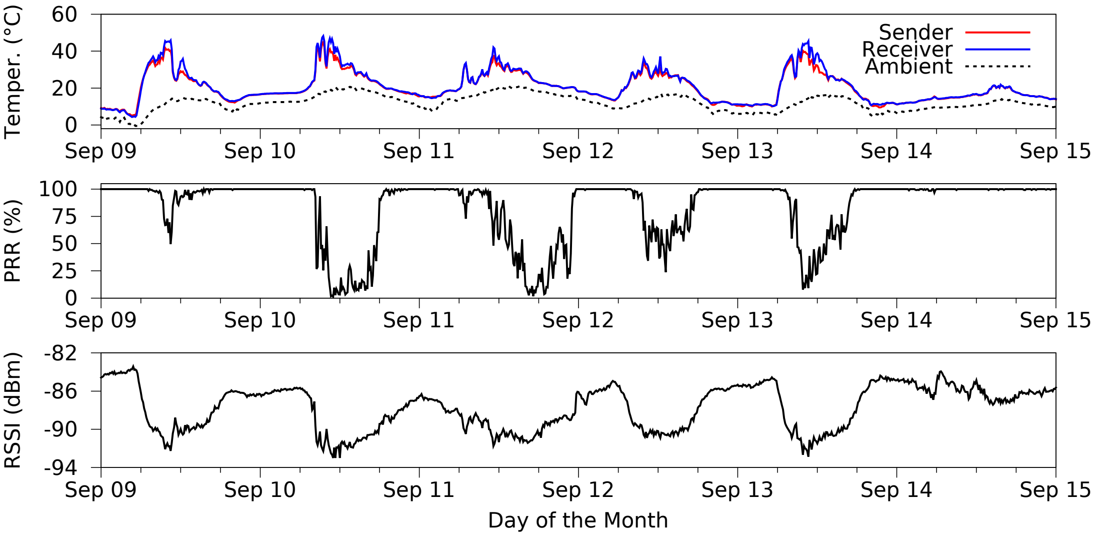


Figure 2.3: Temperature has a strong impact on link quality in outdoor deployments: even the normal temperature fluctuations during a day can render a good link useless. Values are averaged over a timespan of 10 minutes [45].

temperatures, and backed up their controlled experiments with results from an outdoor deployment in the Sonoran desert [23].

Degradation of the wireless link quality. The attenuation of transmitted and received power can strongly affect the quality of links and, ultimately, the reception of packets. Several real-world deployments have indeed shown a correlation between temperature fluctuations and packet reception rate (PRR). The long-term outdoor deployment by Wennerström et al. [220] has shown that packet reception rates are higher during winter than during summer. Whilst in April 80% of good links were sustaining a PRR higher than 90%, in July the number of good links dropped to 35%, and their number increased again during Autumn as soon as the temperatures became milder. In a deployment in an Australian outdoor park, Sun and Cardell-Oliver [198] have measured daily variations of on-board temperature between 10 and 50 °C, showing a completely different performance between day and night on individual links. Different radio performance across day and night was also reported by Lin et al. [134], with daily variation of received signal strength up to 6 dBm, and the highest received signal strength values being recorded during night time.

Although these studies have highlighted a dependency between sensornet performance and temperature, they did not investigate the performance loss further, and did not clarify whether it was actually caused by the impact of temperature on electronics. Some of the aforementioned works indeed argue that communications during night are less prone to radio interference, and present correlations between the PRR and the *air temperature* measured by external sensors or weather stations. Air temperature, however, does not accurately capture the evolution of the *on-board* temperature of the individual sensor nodes over time. The presence of direct sunlight shining on the nodes, as well as the presence of clouds, buildings or obstacles blocking the sun radiation typically creates largely different temperatures variations, as shown in Figure 2.1, and air temperature might hence not clearly correlate with the degradation in link quality.

Analysing the traces recorded by Wennerström et al. in their outdoor deployment, we have found a clearer correlation between the *on-board* temperature fluctuations and the degradation of the wireless link quality.

Figure 2.3 shows the data recorded by two of the nodes deployed in Uppsala during a week in September [220]. One can observe that during daytime (when temperature is high), the received signal strength indicator (RSSI) and link quality indicator (LQI) are lower than during the night. During daytime, also the packet reception rate is reduced, showing that real-world temperature variations can transform a perfect link (100% PRR) into an almost useless one (≈ 0% PRR).

2.2.5 Open research challenges

The research community has undoubtedly produced a large number of solutions to mitigate the impact of temperature on WSN performance. Several clock drift compensation mechanisms have been proposed and validated, as well as battery models capturing the discharge as a function of temperature. Active and leakage power of processors has been measured and characterized in the presence of temperature variations, and variability-aware duty cycling methods have been shown to yield up to 22x improvement in total active time compared to schedules based on worst-case estimations of power [217].

The largest gap that still needs to be filled is a complete understanding of the link quality degradation at high temperatures, and its impact on communication protocols. While the macro-view of the problem is clear (temperature variations affect the efficiency of the radio, degrading the link quality and in turn packet reception), previous research did not formalize the dependency between link quality and temperature in a general fashion, nor did it provide a deeper analysis of the problem at the network level.

Figures 2.1 and 2.2 show that some nodes can be much “hotter” than others: the hot nodes will have a shorter transmission range [23], [44], a larger clock drift [182], whereas the lifetime of the colder nodes will be significantly shorter [159]. How do all these temperature effects affect the operation of network protocols still needs to be answered.

Dependency between link quality and temperature. Currently, every study or deployment report illustrating the dependency between temperature and link quality is unique for experimental setup and hardware employed: the radio chips utilized range from the Nordic NRF903 [198] and the CC1000 [207] to the popular CC1020 [33] and CC2420 transceivers [134], [220]. This makes it difficult to compare the different results and derive insights that can be generalized in a platform-independent analytical model that may be later used to design environment-aware protocols.

Impact of temperature on communication protocols. Communication protocols often rely on signal strength readings and link quality estimation metrics. If and how temperature variations affect their operation has not been thoroughly investigated. For example, MAC protocols often rely on signal strength readings to avoid collisions and to wake-up from low-power mode, and the attenuation of signal strength at high temperatures is likely to harm their operation.

2.3 Impact of Radio Interference on WSN Performance[1]

The massive proliferation of wireless devices in the last decades has caused the radio spectrum to become a very constrained resource. Hence, many standardized technologies operate in increasingly crowded and lightly regulated ISM radio bands. The latter are freely-available portions of the radio spectrum internationally reserved for industrial, scientific and medical purposes.

When several technologies (or multiple devices using the same technology) operate in the same ISM radio band, many devices concurrently share the same frequencies [240]. In such settings, coexistence may become problematic especially for low-power wireless technologies, as the presence of neighbouring devices transmitting at higher power may lead to unpredictable medium access contention times and high latencies, and to a significant increase in the packet loss rate. The latter is typically followed by an increase in the network traffic due to retransmissions, as well as by a decrease in the performance and efficiency of the overall network. Experiences from several wireless sensor network deployments have shown that an unexpected increase of network traffic compared to the initial calculations leads to an early battery depletion or even to a deployment failure [31, 126].

Radio interference can be distinguished in two categories [38]:

- *Internal interference* is the one generated by surrounding sensor nodes operating in the same network.
- *External interference* is caused by wireless appliances operating in the same frequency range of the network of interest using other radio technologies, e.g., Wi-Fi access points and Bluetooth devices.

While internal interference can be minimized by means of a proper configuration of the network (e.g., by a careful placement of nodes) and protocol selection (e.g., by making use of time diversity to avoid concurrent activities in the channel), the mitigation of external interference is often more complex for several reasons. Firstly, it is hardly possible to know in advance all potential sources of interference in a given environment and to predict their behaviour. Secondly, interference is often intermittent and highly dynamic, therefore it is difficult to create solutions that guarantee a reliable communication.

[1] This section is an excerpt of a book chapter reviewing existing work on external radio interference in wireless sensor networks published at Springer in 2013. The chapter was entirely written by the author of this Dissertation under the guidance of his supervisor.

Reference: C.A. Boano and K. Römer. External radio interference. *In "Radio Link Quality Estimation in Low-Power Wireless Networks, SpringerBriefs in Electrical and Computer Engineering – Cooperating Objects"*, pages 21–63. Editors: N. Baccour, A. Koubâa, C.A. Boano, L. Mottola, H. Fotouhi, M. Alves, H. Youssef, M.A. Zúñiga, D. Puccinelli, T. Voigt, K. Römer, and C. Noda. July 2013. Springer International Publishing. DOI: 10.1007/978-3-319-00774-8_2

Link: `http://link.springer.com/chapter/10.1007/978-3-319-00774-8_2`.

Copyright: From the Springer License Terms and Conditions: "With reference to your request to reprint in your thesis material on which Springer Science and Business Media control the copyright, permission is granted, free of charge. [...] This License includes use in an electronic form, provided its password protected or on the University's Intranet or repository."

The remainder of this section focuses on external radio interference. First, interference mitigation solutions that have been proposed in the literature are classified. Thereafter, the yet open challenges are pointed out.

2.3.1 Coexistence between different wireless technologies

The IEEE 802.15.4 standard specifies the two lowest layers of the protocol stack for low-rate wireless personal area networks. The physical layer is responsible for the data transmission and reception according to specific modulation and spreading techniques, as well as for the channel frequency selection and for the management of energy and signal functions (e.g., LQI and energy detection). WSN systems compliant to the IEEE 802.15.4 standard typically operate on one out of three unlicensed ISM frequency bands [38]:

- 868.0-868.6 MHz, available in Europe, one communication channel with center frequency $F_c = 868.3$ MHz;

- 902-928 MHz, available in North America, up to ten communication channels with center frequency $F_c = 906 + 2 \cdot (k - 1)$ MHz, for k = 1, 2, . . . , 10;

- 2400-2483.5 MHz, available worldwide, up to sixteen communication channels with center frequency $F_c = 2405 + 5 \cdot (k - 11)$ MHz, for k = 11, 12, . . . , 26.

Because of the increasing congestion in these ISM bands, several amendments were defined to support new bands, and the standardization process is still evolving nowadays. Among the important updates and amendments made in recent years, two optional layers operating in the 868/915 frequencies employing a different modulation scheme have been added, as well as new physical layers making use of Ultra-wide Band (UWB) and Chirp Spread Spectrum (CSS) modulation techniques.

The 868/915 MHz ISM bands. The 868 and 915 MHz frequency bands are known to be relatively interference-free [226]. Nevertheless, cellular phones can be a significant source of interference for sensor nodes, due to the proximity of the European GSM band [24]. Kusy et al. [124] have also highlighted that telemetry networks and cordless telephones, as well as mobile phones can generate interference in the 900 MHz frequencies. Furthermore, several wireless devices marketed in Europe, including wireless domestic weather stations, car alarms, garage openers, and residential electronic alarms, use the 868 MHz frequency and are therefore potential sources of interference for wireless sensor networks operating in the 868/915 MHz ISM bands.

The 2.4 GHz ISM band. To date, the 2.4 GHz is by far the most congested ISM band, because of the pervasiveness of devices operating in those frequencies, and their high transmission power. Wireless sensor nodes must indeed compete with the communications of Wi-Fi (IEEE 802.11) and Bluetooth (IEEE 802.15.1) devices, as well as with the RF noise generated by microwave ovens and other domestic appliances such as cordless phones, baby monitors, game controllers, wireless presenters, and video-capture devices [161, 189, 240]. The three prevailing sources of interference in this ISM band, i.e., Wi-Fi and Bluetooth devices as well as microwave ovens are shortly described next.

Bluetooth devices. The IEEE 802.15.1 (Bluetooth) standard specifies 79 channels, spaced 1 MHz, in the range 2402-2480 MHz. The interference generated by Bluetooth devices is uniformly distributed across the whole 2.4 GHz band: Bluetooth uses the Frequency Hopping Spread Spectrum (FHSS) technology to combat interference and fading and hops 1600 times per second, i.e., it remains at most 625 μs in the same channel. As of version 1.2, several Bluetooth stack implementations apply an Adaptive Frequency Hopping (AFH) mechanism in which the hopping sequence is modified to avoid interfered channels. Several experimental works have studied the impact of IEEE 802.15.1 communications on the reliability of sensornet transmissions. The packet loss rate of a wireless sensor network operating in the presence of Bluetooth interference typically varies between 3% (as reported by Bertocco et al. [28] and Penna et al. [160]) and 5% (as reported by Huo et al. [104]), up to a maximum of 9-10% (as shown in the experimental results of Sikora and Groza [189]).

Wi-Fi devices. The IEEE 802.11 (Wi-Fi) standard specifies 14 channels, each of which with a bandwidth of 22 MHz, in the range 2400-2483.5 MHz. The coexistence between IEEE 802.11b/g/n and IEEE 802.15.4 devices is extremely challenging due to several aspects. Firstly, Wi-Fi devices can transmit at significantly higher power ($\approx$ 24 dBm) than traditional low-power sensor nodes. Secondly, Wi-Fi devices can use data rates up to 150 Mbit/sec (IEEE 802.11n standard) and are ubiquitous nowadays, especially in residential and office buildings. Furthermore, the large bandwidth of Wi-Fi channels allows them to interfere with multiple IEEE 802.15.4 channels at the same time. Several experimental works have investigated the impact of IEEE 802.11 communications on the reliability of IEEE 802.15.4 transmissions [28, 97, 132, 160, 161, 189]. Their results show that concurrent Wi-Fi transmissions can block the majority of WSN transmissions (up to 90% loss) and cause long delays, drastically decreasing the performance of the network.

Microwave ovens. Several studies have highlighted that microwave ovens can be an important source of interference in the 2.4 GHz ISM band [114, 200, 216]. Frequency-wise, microwave ovens can interfere a large portion of IEEE 802.15.4 channels at a very high power (they operate at up to 60 dBm). The interfered channels vary with changes in load impedance, supply current, and temperature of the magnetron [216]. The latter is affected by the amount of water in the food and the position of the latter within the oven, making it hard to foresee the disturbed frequencies. Time-wise, the interference generated by microwave ovens is rigorously periodical, as ovens continuously turn on and off according to the frequency of the AC supply line. Hence, the duration of a power cycle depends on the power grid frequency (roughly 20 ms at 50 Hz, and 16 ms at 60 Hz).

2.3.2 Classification of interference mitigation techniques

A large number of works studied the coexistence problem in wireless sensor networks and proposed several interference mitigation techniques [38, 234]. The latter can be classified in five main classes: frequency diversity, space diversity, hardware diversity, time diversity, and redundancy, as shown in Figure 2.4.

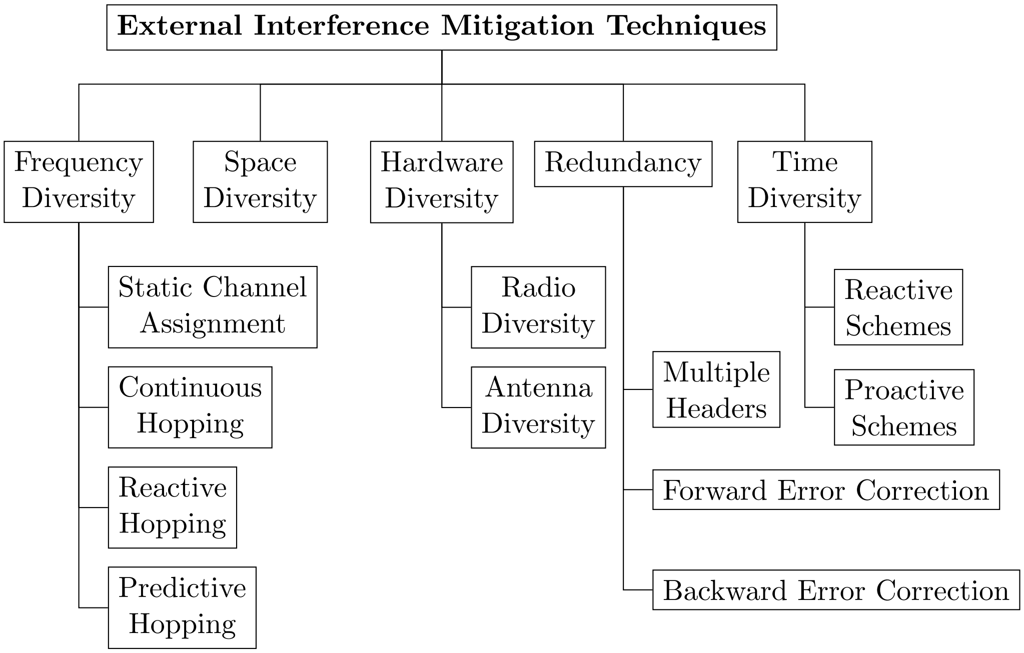


Figure 2.4: Taxonomy of existing techniques to mitigate external interference [38].

Frequency diversity. Techniques exploiting the availability of multiple radio channels in a given ISM band to avoid external interference can be divided in four sub-categories.

Static channel assignment. A primordial way to pursue interference avoidance consists in statically assigning the frequency channels depending on the expected interference sources, i.e., each node gets assigned to a fixed IEEE 802.15.4 channel that is supposedly interference-free. For example, several networks operate on channel 26 in order to escape Wi-Fi interference [60, 131]. Although channel 26 typically offers the best performance, this strategy is not optimal, as there may be co-located networks following the same assumption, or unexpected interfering devices such as microwave ovens occasionally operating in the surroundings. The same disadvantage applies to a number of works that statically assign portions of the network to specific channels [120, 157, 178, 191, 228, 229]. These multichannel protocols can indeed maximize the bandwidth available for communications by increasing the number of channels, but do not enhance or guarantee coexistence among devices operating in the same frequency range.

Continuous hopping. Another way to pursue interference avoidance resembles the FHSS technique of the IEEE 802.15.1 standard, and consists in continuously hopping among channels according to the same pseudo-random sequence. Hopping can be blind (i.e., nodes hop among all available channels) or adaptive, i.e., nodes carry out some form of blacklisting of undesirable congested channels [219]. Examples of protocols adopting adaptive continuous hopping are the Time Synchronized Mesh Protocol [162], the WirelessHART standard [193], the protocols developed by Du et al. [70] and Yoon et al. [236], as well as EM-MAC [202]. The latter uses a penalty system with channel blacklisting based on the results of the clear channel assessment (CCA) operation. A node switches among channels based on its pseudo-random channel schedule, except that, if the next pseudo-

randomly chosen channel is on the node's channel blacklist, the node stays on its current channel.
Continuous hopping exploits the potential of frequency diversity and hence can reduce the impact of narrow-band interference and persistent multipath fading. Furthermore, channel hopping also ensures fairness among the chosen channels. However, channel hopping requires a tight time synchronization across the network in order to work properly. Also, the seed and the list of blacklisted channels needs to be shared in a reliable fashion, a critical operation in the presence of interference. It is important to note that in case of blind continuous channel hopping, the interference avoidance is only passive, i.e., by continuously hopping, a pair of nodes will sooner or later pick a good communication channel. This, however, might not necessarily happen in a short time interval. Another drawback of continuous channel hopping is the energy required to continuously switch channels. Compared to protocols switching on demand, this represents a non-negligible burden. Also, adaptive continuous hopping protocols require to continuously update and spread the list of blacklisted channels, which may require a significant amount of energy.

Reactive hopping. In order to avoid the burden of continuously switching the channel, several protocols switch or blacklist channels only once performance has degraded, e.g., in only a specific part of the network [227]. These approaches continuously monitor the quality of the current channel and check whether it is satisfactory: if too much interference is detected, a frequency switch is carried out. These protocols are *reactive* in the sense that to mitigate interference by frequency diversity, they first need to experience a performance degradation. In this category fall protocols such as CoReDac [215], Chrisso [105], and ARCH [185]. The latter uses the expected number of transmissions (ETX) to monitor the quality of the link, and as soon as the ETX values collected in a given observation window exceed a certain threshold, a new channel is selected. The authors show that 15 minutes of observation are enough to predict channel reliability, and a notable method is suggested to select the next channel. After blacklisting the current channel, ARCH hops to a new channel that is further away from the previous one. This has two benefits: on the one hand it avoids wideband interferers, on the other hand it avoids deep fades, as highlighted by Watteyne et al. [218].
The advantages of reactive protocols are, as mentioned above, the significant energy saving compared to hopping continuously and maintaining time synchronization among nodes. However, such protocols may not suit safety-critical systems, as they need to experience packet loss before performing a channel switch. Moreover, the switch is often performed blindly, i.e., without necessarily knowing the stability of the other channels.

Proactive hopping. A few works try to avoid packet loss by predicting when the channel conditions will degrade and by hopping before this happens. A fundamental role in the development of proactive protocols is played by channel quality estimation metrics that can detect an early degradation of the channel, e.g., [97, 145], as well as by an efficient link quality ranking algorithm, such as [244]. The work by Kerkez et al. [118] keeps track of the quality of all channels by periodically forcing a channel switch and selects the more reliable channels. In MuZi [231] all the channels are scanned and a new reliable channel is selected (but only as soon as the performance of a channel has degraded). The main disadvantage of these approaches is that they heavily rely on an accurate channel quality estimation (based on energy detection), which can be very costly in terms of energy consumption.

Space diversity. A solution widely investigated in the context of large and dense wireless sensor networks consists in avoiding interference by routing packets through different links. Adaptive routing has been studied by several works that do not explicitly target the presence of external interference, but rather aim for an effective link estimation in order to achieve reliable communications. Alizai et al. [10] proposed to apply a bursty routing extension to detect short-term reliable links. Their approach allows a routing protocol to forward packets over long-range bursty links in order to minimize the number of transmissions in the network. Liu and Cerpa [136] have developed a receiver-driven estimator based on a machine learning approach to predict the short temporal quality of a link. Their estimation is based on trained models that predict the link quality using both packet reception rate and other physical layer parameters, such as RSSI, SNR, and LQI. Gonga et al. [89] have carried out a comparison between multichannel communication and adaptive routing, in order to determine which one guarantees more reliable communications in the presence of external interference and high link dynamics. The authors have shown that adaptive routing performs well in dense wireless sensor networks. The key reason behind this is the selection of good long-term stable links, which avoids low-quality links that may be more vulnerable to external interference. When external interference is present, one could indeed try to route a packet towards a closer node, so that the probability that a stronger signal corrupts the packet is smaller. However, in sparse networks, where the choice of forwarding links is rather limited, adaptive routing looses its flexibility, and multichannel solutions yield better performance in terms of both average end-to-end delay and reliability.

Hardware diversity. Several works have made use of wireless sensor network platforms equipped with dual radios to communicate in multiple ISM bands. Other works have proposed the use of directional antennas or spatially separated antennas to achieve more reliable communications even in the presence of interference.

Radio diversity. Kusy et al. [124] have shown that radio transceivers operating at different radio frequencies and through spatially separated antennas offer robust communication, high link diversity, and better interference mitigation. Using dual radios, the authors have experimentally shown a significant improvement in the end-to-end delivery rates and network stability, at the price of a slight increase in energy cost compared to a single radio approach. Examples of wireless sensor network platforms equipped with dual radios are the BTnode, the Mulle node, and the Opal node.

Antenna diversity. Rehmani et al. [170] have envisioned the possibility to design and implement a software-defined intelligent antenna switching capability for wireless sensor nodes. More precisely, the authors have attached an inverted-F antenna to a TelosB mote in addition to the built-in antenna in order to achieve antenna diversity. Based on the wireless link condition, and in particular on physical layer measurements, the sensor node should then dynamically switch to the most appropriate antenna for communication. Another option are dynamically steerable directional antennas [87]. The latter are able to dynamically control the gain as a function of direction, and, because of these properties, they can be very useful to increase the communication range and reducing the contention on the wireless medium. In [87], the authors have proposed a four-beam patch antenna and showed interference suppression from IEEE 802.11g systems. They have further discussed the use of the antenna as a form of angular diversity useful to cope with the variability of the radio signal. Further examples of directional antennas used in WSN systems in which a sensor node can easily concentrate the transmitted power towards the intended receiver dynamically can be found in [148, 156, 213].

Redundancy. Several solutions exploit redundancy to mitigate the impact of external interference, such as the use of multiple headers, as well as forward and backward error correction techniques.

Multiple headers. Liang et al. have experimentally shown that IEEE 802.11 transmitters can back off due to elevated channel energy when nearby IEEE 802.15.4 nodes start sending packets [131, 132]. When this happens, the IEEE 802.15.4 packet header is often corrupted, but the rest of the packet is still intact. Based on this observation, the authors have proposed the use of multi-headers (as in Bluetooth) to protect the IEEE 802.15.4 packets from the corruption generated by Wi-Fi interference. The authors have suggested that two additional headers represent a good trade-off between overhead and performance. It is important to notice that multi-headers are only effective in the so called symmetric region, i.e., when an IEEE 802.15.4 transmission is able to affect the behaviour of an IEEE 802.11 transmitter, because the bit errors occur mainly in the beginning of the packet. In contrast, in the asymmetric region, i.e., when the IEEE 802.15.4 signal is too weak to affect IEEE 802.11 behaviour, the bit errors are distributed across the packet in a uniform way.

Forward error correction techniques. In order to mitigate interference in the asymmetric region, Liang et al. [131, 132] have also proposed the use of forward error correction (FEC) techniques to recover from corrupted packets. When using forward error correction, the original message is encoded into a larger message by using an error correction code, which implies a longer time in which the radio is switched on, and a longer computation time for encoding and decoding the packet. The receiver then decodes the original message by applying the reverse transformation of the error correction code. The redundancy in the encoded message allows the receiver to recover the original message in the presence of a limited number of bit errors. The authors demonstrated that Reed-Solomon (RS) correcting codes perform well while recovering packets corrupted by IEEE 802.11 interference [132]. However, FEC techniques pose a trade-off between data recovery capacity and its inherent payload and computation overhead. Forward error correction indeed creates overhead both on the receiver and the transmitter, and therefore requires a significant amount of energy as well as noteworthy computational capabilities. Liang et al. [132] have shown that the time required to encode an original 65-byte message into an RS-encoded message with 30-byte parity is approximately 36 milliseconds, whereas the decoding of the message depends on the presence of errors and can vary between 100 and 200 milliseconds.

Backward error correction techniques. An alternative to forward error correction is the use of acknowledgement (ACK) or negative-acknowledgement (NACK) packets to trigger a retransmission of the corrupted frames. This solution may not necessarily lead to a good result in the presence of external interference, as retransmitted packets are prone to corruption as well as the original packet. Furthermore, when sending ACK or NACK packets, one may increase the channel congestion and the energy consumption of the motes. In order to minimize the energy consumption required for retransmissions, Hauer et al. [98] have developed an Automatic Repeat reQuest (ARQ) scheme that minimizes the amount of data that the sender needs to retransmit. In their scheme, the receiving node triggers only the retransmission of the damaged portion of the packet, which saves a significant amount of energy in case the packets had a long data payload. Receivers record the RSSI of the received packet during reception at high frequency, and try to estimate the position of the error within a packet (if any). This RSSI-based recovery mechanisms is effective also in the presence of external interference, because collisions of frames with the transmissions generated by other devices such as IEEE 802.11 or IEEE 802.15.4 can be detected through an increase in the RSSI profile, which would otherwise be very stable (typically ± 1 dBm).

Time diversity. Another class of external interference mitigation techniques is time diversity, which consists in either deferring transmissions, or scheduling them in such a way to avoid interference.

Reactive schemes. A basic way to mitigate interference is to defer transmission until interference clears, using for example the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) technique. However, depending on the congestion back-off scheme and CCA threshold selected, the latency can drastically increase in the presence of interference. Yuan et al. [237] have proposed a decentralized approach in which sensor nodes adaptively and distributively adjust their CCA thresholds, and have shown that this approach substantially reduces the amount of discarded packets due to channel access failures, and hence increases the performance of sensornet protocols under interference. Bertocco et al. [27] study the performance of different CCA modes in the presence of in-channel wide-band additive white Gaussian noise. Similarly, Petrova et al. [161] investigate the three CCA modes defined by the IEEE 802.15.4 standard (energy above threshold, carrier sense only, and carrier sense with energy above threshold). They have observed that dynamic CCA thresholds can improve the performance of sensornet communications both in the overlapping and non-overlapping channels with IEEE 802.11n.

Proactive schemes. Another class of protocols is the one in which the sensor node tries not to defer transmissions, but rather to schedule them in a way to avoid the interference of other devices. An example from this class is the scheduling of packets proposed by Chowdury and Akyildiz [61]. The authors have analysed the cases in which microwave ovens and Wi-Fi device are operating, and proposed a scheme in which the sensor nodes transmit whenever the channel is predicted to be free based on the Wi-Fi traffic or microwave oven duty cycle. In the case of microwave oven interference, the sensor nodes can align their own sleep cycles with the duty cycle of the microwave oven, and synchronize their transmissions with the beginning of the off-time. In the case of Wi-Fi transmissions, the sensor nodes exploit the detection of Short Inter-Frame Space (SIFS) and Distributed Inter-Frame Space (DIFS).

2.3.3 Open research challenges

The research community has produced a substantial number of solutions addressing the problem of radio interference for low-power wireless communications. Previous research has mostly focused on (i) highlighting the coexistence problem in congested ISM bands and on (ii) designing techniques that maximize the probability of packet reception under interference. There are, however, a number of gaps that still need to be filled.

Comparison of interference mitigation techniques. Most of the mechanisms presented in the earlier section were not compared against each other in the presence of different interference patterns. This does not clarify the advantages and disadvantages of a solution against another, and hinders the integration in state-of-the-art WSN protocols. For example, several of the proposed solutions are tailored to specific interference patterns (e.g., Wi-Fi [132] and microwave ovens [61]), but it is not clear how they would perform in the presence of other interfering devices. Ideally, there should be a guide for system designers and developers containing hints about which of the available protocols are more suitable in a given environment. One of the fundamental issues towards this goal is the lack of low-cost WSN testbed infrastructures enabling the repeatable playback of specific environmental conditions, as discussed in Section 2.4.1. Most sensornet testbeds are indeed deployed in indoor office environments and, although the latter may be rich of Wi-Fi devices in the surroundings, there is no simple way to control the interference that is generated and obtain sound performance comparisons. Furthermore, experimentation is constrained to the interfering devices available in proximity of the nodes, which limits the diversity of the interference patterns against which protocols are tested.

Characterization of protocol performance under interference. Communication protocols typically have configurable parameters that can change their behaviour or increase their efficiency in specific settings. A precise characterization of protocol performance as a function of these configurable parameters under interference is still missing, and would be extremely helpful to identify (i) which configurations break protocol operation in the presence of a congested channel and (ii) how these parameters should be configured to maximize performance in the presence of specific interference patterns. For example, at the MAC layer, the choice of a number of parameters such as the congestion back-off scheme, the clear channel assessment threshold, as well as the duration of the strobing period, may play a fundamental role in the presence of interference, and their selection could be adjusted in order to offer better performance.

Agreement under interference. One aspect that has not been tackled in detail is how interference affects the exchange of fundamental pieces of information between two nodes. A critical building block of many protocols at all layers is indeed the *agreement* on a piece of information among a set of nodes. At the MAC layer, nodes may need to agree on a new time slot or frequency channel; at the application layer nodes may need to agree on handing over a leader role from one node to another [7]. Message loss may break agreement in two different ways: none of the nodes uses the new information (time slot, channel, leader) and sticks with the previous assignment, or – even worse – some nodes use the new information and some do not. This may lead to reduced performance or failures. Although several of the techniques proposed in the previous section can be used to maximize the probability of reception in the presence of interference (e.g., time diversity and redundancy), nodes typically do not have a guarantee that a packet has been correctly received, as interference can also destroy acknowledgement messages transmitted as part of a handshake to verify the correct reception of packets. Hence, this problem requires attention, as disagreements may lead to reduced performance or complete network failure.

Lightweight interference models. Another open challenge is the creation of accurate models of common interference sources that can be exploited by resource-constrained sensors nodes. Ideally, a lightweight interference model would enable even sensor nodes with severe hardware limitations and a reduced energy budget to capture the characteristics of interference in the surroundings at runtime. This could for example allow a node to identify the interfering devices operating in their proximity and to select the most appropriate interference mitigation technique. Similarly, lightweight interference models would allow a node to carry out runtime adaptation of protocol parameters.

2.4 Experimentation in WSN Testbeds

The wireless sensor networks research community traditionally relies on testbeds to evaluate and optimize communication protocols and applications under realistic conditions in a cost-effective way. Testbed infrastructures are a powerful tool for validation and performance evaluation of algorithms and protocols, as they offer the possibility to quickly upload software on a relatively large network scale and easily retrieve the collected measurements. Testbeds complement simulation environments by enabling experimentation on tangible hardware and in real-world settings, with minimal set-up time and easy maintenance. Simulation tools are instead mainly based on mathematical models that do not fully capture the complexity of the real-world, and often ignore the interactions between communication protocols and underlying hardware platforms.

A large number of WSN testbed facilities has been developed in the last decade (the vast majority of which is deployed at academic institutions), and differ by number of available nodes, supported hardware and software, availability to the general public, network management software, as well as scheduling, management software, and user interface [195]. The most renowned testbeds are publicly available, i.e., registered users can upload the specifications of an experiment and collect traces directly via a web interface. Examples are MoteLab [224], one of the first open-source wireless sensor networks testbeds to be developed (and still one of the largest, with its 190 nodes deployed over 3 floors), Kansei [78] (210 sensor nodes with a gateway station attached to each of the sensor nodes), Indriya [67] (127 TelosB nodes deployed at the National University of Singapore), TWIST [93] (200 heterogeneous nodes across several floors in a building in the campus of the Technical University of Berlin, Germany), and NetEye [113] (130 TelosB mote deployed at at Wayne State University, MI, USA).

Sensornet testbeds have constantly evolved in the last years to provide a better service. Focus has been on reducing their management effort [63], allocating testbed resources to users that need them the most [62], accurately analysing the power consumption [94], improving data presentation and analysis [65], testing the same experiment in different locations or at different testbed installations [171], as well as on the confederation of multiple testbeds (notable examples are WISEBED [59, 99], Senslab [174], and X-Sensor [116]). To simplify migration between simulation and testbed, checkpointing of system state has been proposed [154], e.g., to import into simulation a realistic topology or network state that occurred during the experimentation in testbeds. To allow users to control software executions with fine-grained profiling and tracing, Aveksha [201] provides an enhanced visibility into the internal state of the processor and three modes of event logging and tracing, whereas Envirolog [138] provides repeatability of executions based on scoped event readings. Baccour et al. [19] have developed RadiaLE, a testbed in which nodes are deployed in a radial topology, that provides the ability of automating the evaluation of link quality estimators by analysing their statistical properties. A few testbeds also give the possibility of emulating sensor data by feeding the testbed with data gathered in live experiments to allow a convenient and repeatable testing. The T301 testbed [204], for example, provides a sensor emulation feature that allows the user to import generated sensor data or traces gathered in live experiments.

2.4.1 Environmental control in WSN testbeds

The significance of a testbed experiment largely depends on how accurately environmental effects can be reproduced. Therefore, testbed infrastructures should provide the ability of repeating the same experiment under the same environmental conditions, to allow, for example, to investigate how protocol performance is affected by a certain parameter.

Recent works have extended existing testbed infrastructures with the emulation of node mobility. In ViMobiO [168], Puccinelli and Giordano implemented a virtual mobility overlay to reproduce movement patterns of nodes during experimental evaluation. In the CONET testbed [1], a swarm of five Pioneer 3-AT autonomous robots communicates with a static wireless sensor network [110, 111]. Similarly, in Mobile Emulab [112], several robots carrying Mica2 nodes can move in a $60\,m^2$ space, and the testbed provides simple path planning as well as vision-based tracking system, plus the possibility of monitoring the experiment through web-cams.

Recent efforts have also extended existing infrastructures with the emulation of radio interference. Slipp et al. [190] have developed WINTeR, a testbed facility to support implementation and evaluation of wireless sensor networks for industrial settings. WINTeR allows to replicate the harsh RF conditions in industrial environments by means of an Anritsu MG3700A VSG-based EMI generator operating in the frequency range [250 kHz - 6 GHz]. Sanchez et al. [181] have envisioned a novel testbed federation incorporating SDR devices, which would facilitate recording and playback of interference patterns. To allow research on advanced spectrum sensing and cognitive networking strategies, several testbed facilities are being augmented with state-of-the-art cognitive systems [5].

2.4.2 Open research challenges

Despite testbed capabilities having significantly evolved in the past years, the ability of reproducing different environmental conditions is, to date, still rather limited, especially with respect to the playback of factors like temperature and radio interference. Several of the aforementioned approaches to generate realistic interference patterns involve rather expensive equipment: the cost of SDR hardware and VSG-based EMI generators is indeed still very high, and these approaches would not scale to large testbeds. A low-cost testbed infrastructure in which realistic interference patterns can be created in a quick, simple, yet accurate fashion, would allow several researchers around the world to compare their solutions, and to advance the state-of-the-art in a much faster way. Similarly, testbed facilities allowing to vary the on-board temperature of each sensor mote individually would facilitate more accurate studies of the impact of temperature variations on network performance.

Efficient experimentation with radio interference. The lack of low-cost testbed infrastructures in which realistic interference patterns can be created in a quick, simple, yet accurate fashion, is one of the reasons why experimentation with radio interference is, to date, still cumbersome and labour-intensive. Researchers and developers indeed rarely resort to simulation tools, as mathematical models such as the unit disk model and the log-normal shadowing model do not capture the behaviour of wireless sensor nodes with a sufficiently high accuracy [84]. In fact, wireless propagation strongly depends on the hardware used, on antenna irregularities, on the geometry and nature (static or mobile) of the environment, as well as on shadowing and multipath fading, which are extremely complex to model.

A large number of researchers exploit the ambient interference in the surrounding of wireless sensor nodes to evaluate protocols and applications [38]. This approach consists of running experiments in environments rich of co-located Wi-Fi access points, such as office environments [70, 89, 152], university campuses [88, 98, 239], libraries [149], and residential buildings [186]. Exploiting noisy environments has two main advantages. First, the interference patterns against which protocols and applications are tested are real. Second, the costs and efforts required to set up the experiments are minimal. This, however, comes at the price of complete uncontrollability and non-repeatability of the experiments. Indeed, one does not have complete knowledge of the devices that are actually operating in the surroundings, nor can control them. Also, comparability among experiments becomes very complex, and experimental results should be taken “with a grain of salt”, as interference may drastically change across multiple trials. For example, the interference generated using Wi-Fi devices depends on the frequency selected by the access points at a given time, on the activities of the users, as well as on the congestion of the backbone [38].

Other works explicitly generate interference patterns to evaluate protocols and applications [38]. Some researchers exploit the presence of nearby IEEE 802.11 access points to trigger file transfers [75, 115, 127], and to continuously transmit UDP packets at different rates [132, 142, 145]. An alternative consists in creating custom experiments by manually placing interfering devices in proximity of wireless sensor nodes. These kinds of experiments are very popular, as they are cost- and time-effective, but are typically small-scale and only involve a small number of sensor nodes. Furthermore, there is a lack of device diversity: the vast majority of works uses only IEEE 802.11b/g [9, 14, 102, 103, 109, 188] and IEEE 802.15 devices [16, 28, 104, 160]. Only a few studies involve more advanced equipment, such as signal generators, and software-defined radios [13, 169], allowing higher degrees of controllability.

A few works use sensor nodes to produce interference [215, 232] by continuously transmitting packets using the IEEE 802.15.4 channel of interest. Although it requires only a limited setup time and no additional hardware is required, the main drawback of this approach is that the transmission of packets is not fully controllable (inter-packet times are not programmable directly [35]) and that the interference does not resemble the one produced by other devices, as it is constrained by the size of the transmitted packets.

Temperature control in WSN testbeds. Industry makes often use of temperature chambers during device verification processes (e.g., to calibrate sensors and transceivers [58]), but such solutions are not suitable to understand the impact of temperature on WSN performance.

First, temperature chambers can be very expensive. Second, they target individual components and not *a network* of nodes, which is necessary to disclose limitations at the communication level. Experimenting inside thermal chambers targets indeed only individual components and not a network of nodes (with different individual temperatures), which is necessary to disclose limitations at the network level. Furthermore, there would be implications on the propagation of signals due to the metal casing when carrying out experiments using multiple thermal chambers.

A number of researchers have installed outdoor testbed facilities in the past years. A few outdoor solar-powered nodes are currently available at the University of Braunschweig, Germany, as part of the WISEBED confederation [99], whereas in FlockLab [133] four nodes are deployed on top of a roof in Zürich, Switzerland. The Trio testbed [76], deployed in 2005, was one of the largest outdoor wireless sensor network testbeds ever built and consisted of 557 solar-powered motes. However, Trio was never open to the public research community, as well as the experimental solar-powered “CampusNet” testbed deployment [92]. Although experiments on networks deployed outdoors can show the impact of temperature on WSN performance [220], experimentation on outdoor testbeds does not allow a systematic analysis. On the one hand, meteorological conditions cannot be controlled, making it impossible to ensure repeatability across several experiments. Furthermore, the temperature profiles that can be tested are highly specific to the deployment location and to the time of the year in which the experiment is carried out.

Therefore, there is no practical extension for WSN testbeds that allows to vary the on-board temperature of each sensor mote and accurately study the impact of temperature variations on WSN performance on a network level, making experimentation complex. In several works, the temperature variations are triggered manually on two or three devices, for example by means of hot-air guns [57], by leaving motes on electric heaters and moving them outdoors to cool down [241], or by moving nodes in-and outside a refrigerator [24, 33], which does not allow an accurate temperature control on a large scale and hinders repeatability of an experiment.

Chapter 3

Temperature

This chapter describes the contributions of this thesis with respect to the impact of temperature on WSN performance. Section 3.1 describes TempLab, an extension for wireless sensor network testbeds that allows to control the on-board temperature of sensor nodes in a precise and repeatable fashion. TempLab facilitates experimentation with temperature and serves as a tool to investigate the effects of temperature variations on network performance. Section 3.2 analyses the impact of temperature on wireless link quality and derives a platform-independent protocol model that characterizes the attenuation of signal strength on low-power radios. Section 3.3 analyses the impact of temperature on protocol performance, and specifically on state-of-the-art sensornet MAC and routing protocols. Building upon the lessons learnt, Section 3.4 illustrates two mechanisms to dynamically adapt the clear channel assessment threshold to temperature changes and shows how these mechanisms make data link layer protocols temperature-aware, actually mitigating the adverse effects of temperature on protocol performance and increasing the dependability of WSNs.

3.1 TempLab: A Temperature-Controlled WSN Testbed

To better study the impact of temperature variations on low-power wireless communications and protocols, we have designed TempLab, a testbed infrastructure with the ability of varying the on-board temperature of sensor nodes and reproducing the temperature fluctuations that can be normally found in outdoor deployments [45].

Such a testbed solution essentially has one main functional module: the ability to control the on-board temperature of wireless sensor nodes. However, in order to accurately reproduce the temperature dynamics that can be found in typical deployments, it is not simply enough to choose off-the-shelf heating and cooling elements and connect them to a testbed. The choice of the hardware, as well as the design of the infrastructure has been driven by a number of requirements that we describe in Section 3.1.1.

3.1.1 Requirements

In order to faithfully reproduce conditions that can be found in real-world deployments and to support a wide range of experimentation techniques, a temperature-controlled WSN testbed should be able to satisfy the following requirements:

- *Large temperature range.* Ideally, the testbed should be able to reproduce temperature patterns covering the complete operating range of sensor nodes ([-45,85]°C in the case of TelosB-based platforms). However, to reproduce the conditions that can be found in a real deployment, this is not strictly necessary, as long as the testbed can emulate the variations that can be found during the coldest and hottest times of the year in the setting of interest.

- *Fine-grained temperature control.* The temperature of a node deployed outdoors can continuously vary depending on the presence of sunshine and obstacles (e.g., clouds or buildings), causing continuum gradients of temperature. Hence, a fundamental ability of the testbed infrastructure should be the ability of precisely tune the on-board temperature of a sensor node with a high resolution.

- *Fast temperature variations.* In a real deployment, temperature can change quickly due to meteorological effects such as wind, rain, and snow, as well as due to the presence of clouds or sunshine. An important requirement is the ability of reproducing these variations as fast at they occur in the real-world: for example in the deployment shown in Figure 2.1, a node that receives the first sun-rays at the beginning of the day experiences an increase of its on-board temperature up to 1.98 °C/minute.

- *Time scaling.* It is often desirable to compress the time scale of an experiment to save evaluation time (as long as the behaviour of a platform does not depend on the rate of the temperature change, but only on the absolute temperature values). Indeed, one may want to time-lapse the recreation of real-world traces and playback, for instance, in a few hours the profile of a full day. This poses stronger requirements on the ability of the testbed to quickly heat up and cool down nodes.

- *Per-node temperature control.* As discussed in Section 2.2.1, the temperature profiles of each node can be highly different. Placing all the nodes into a single chamber would not be realistic because all nodes would follow the same temperature profile, and temperature must hence be controlled individually on each node.

- *Repeatability.* When comparing or debugging the performance of protocols, it is fundamental to be able to repeat an experiment under the same conditions, i.e., the testbed should reproduce the same temperature profiles across multiple experiments.

- *Unaltered system behaviour.* The extension of the existing infrastructure should ideally not alter the behaviour of the system in any way, as this may lead to unwanted (and unexpected) system failures. For example, the use of metal casings should be restrained, as RF propagation should be minimally affected. Similarly, the use of I/O ports of a sensor node to control heating or cooling devices has to be avoided if this would affect the normal operations of the system.

- *Heterogeneity and scalability.* Although it may not be necessary to augment all nodes of an existing infrastructure with temperature control, it should be possible to extend an entire testbed regardless of the type of sensor nodes used.

- *Low cost.* All the above requirements have to be satisfied while minimizing the cost of the solution, in order to make it applicable on a large scale.

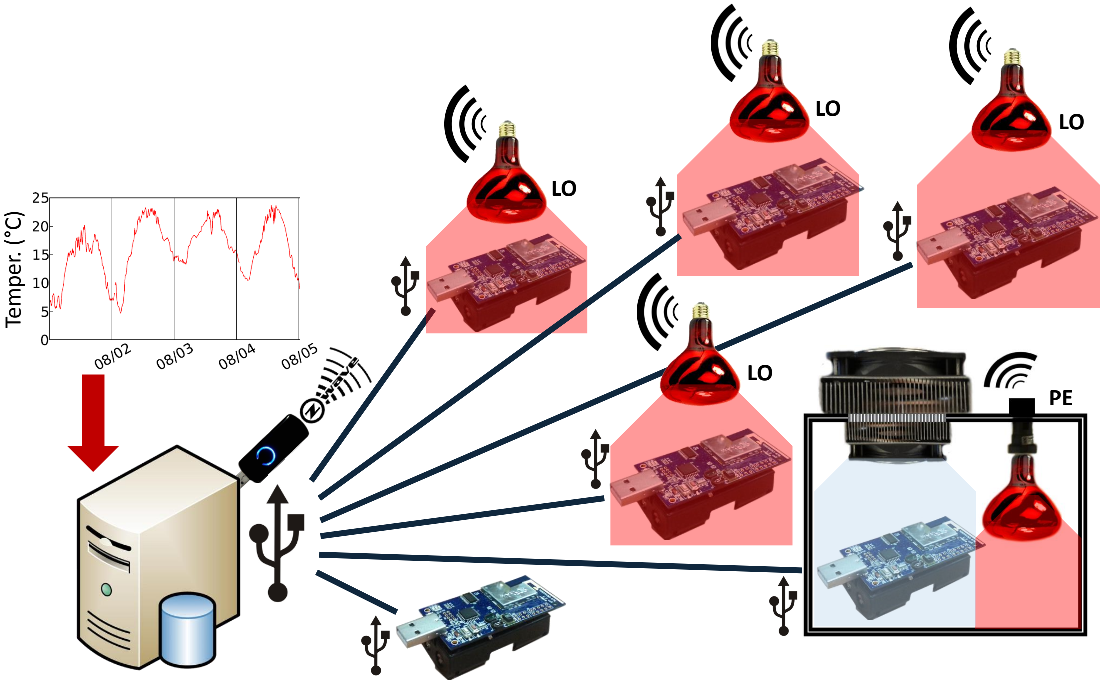


Figure 3.1: Overview of TempLab's architecture [45].

3.1.2 Architecture

TempLab has been designed to meet all the aforementioned requirements by following an out-of-band approach (i.e., sensor nodes are not involved in the control of their temperature, and additional processing hardware is hence needed) based on IR heating lamps and cooling enclosures[1]. An overview of TempLab's architecture is shown in Figure 3.1.

Actuators. TempLab requires actuators to control the on-board temperature of each sensor node, and can support two types of nodes with different capabilities:

- *LO nodes*, which stands for lamps-only nodes, are heating-only devices that have the capability of warming the sensor nodes between room temperature and their maximum operating range. They are based on IR heating lamps, and they do not have any capability to cool-down the nodes below room temperature (an example of LO node is shown in Figure 3.2(b)).

- *PE nodes*, which stands for Peltier Enclosure nodes, are hard temperature-isolating Polystyrene enclosures with an embedded IR heating lamp and an air-to-air Peltier module to heat-up and cool-down the temperature inside of the casing. The selection of polystyrene-based materials (for which RF absorption is minimal [177]) has been driven by the requirement of minimizing the impact of casing on signal strength.

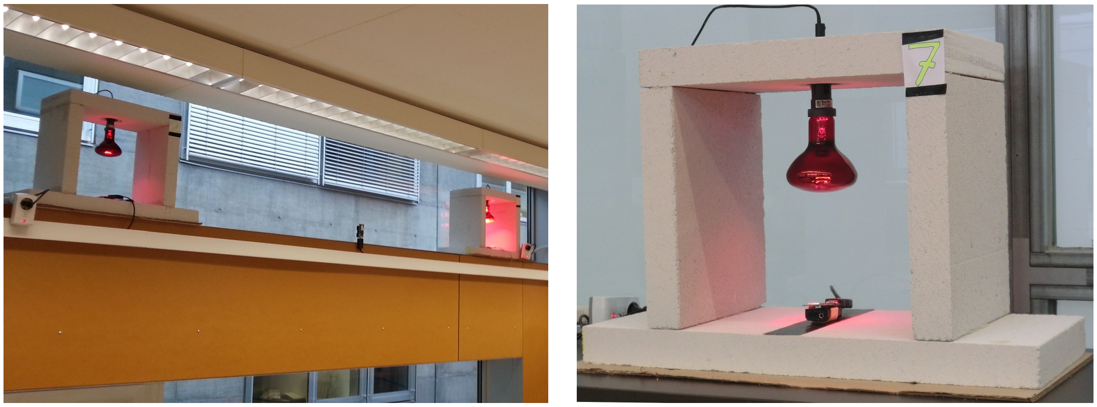
[1] Full detail about TempLab's design and implementation can be found in Paper F included in this thesis [45].

To control the intensity of the IR lamps and the operations of the Peltier module, TempLab borrows existing home automation solutions and uses *wireless dimmers* to vary the intensity of the lamps and *on-off wireless switches* to control the Peltier modules embedded in the enclosure. To make sure that the temperature control system does not interfere with the existing testbed communication, we select home automation solutions working on a ISM frequency band that is different from the one used by the sensor nodes.

This approach can easily scale to large testbeds as PE and LO nodes only need to be plugged into wall outlets and require no further cabling. Furthermore, home automation solutions such as Z-Wave [68] allow multiple devices (LO and PE lamps in our case) to communicate in a multi-hop fashion, and create different home networks each of which can have a maximum of 256 nodes. If a very large number of nodes needs to be supported, it is possible to partition the control network and use several controllers. All what is needed is the availability of a power line, but as in most indoor testbeds there exists a wired back-link to each node, the efforts to add a power line are typically not too high.

Controllers. To instantiate a temperature profile and control heat lamps and Peltier modules, TempLab uses different controllers running on a centralized testbed gateway computer. The simplest one is an *open-loop controller* that varies the intensity of the light bulbs in LO and PE nodes according to a pre-computed calibration function. To precisely regenerate trace- or model-based temperature profiles, TempLab uses a *closed-loop proportional-integral (PI) controller* that tries to minimize the difference between the desired temperature profile and the on-board temperature of the sensor node of interest. The controller should hence receive a periodic feedback with frequency F_U about the on-board temperature of the sensor node in order to minimize the error with respect to the desired temperature profile. As most off-the-shelf wireless sensor nodes carry an on-board temperature sensor, TempLab uses the sensor node itself to measure the temperature and exploits the USB back-channel to periodically convey temperature readings to the controller using a low-priority routine executing only when the processor is idle.

Supported temperature profiles. In order to support a wide range of experimentation techniques, TempLab can generate temperature profiles using three different approaches. Firstly, one can re-play temperature traces collected in-situ at a given deployment site. Such *trace-based* temperature profile instantiation can accurately reflect the temperature variations over time with fine granularity if long-term measurements from one or more nodes are available. Given that traces are not always at one's disposal, a second possibility is to use a *model-based* temperature profile to have an estimation about the temperature dynamics at a certain location without the need of traces collected in-situ. A model-based approach uses models to estimate the temperature profile of objects using basic environmental information such as the maximum solar radiation and the minimum temperature during a day (that is readily available from satellites and meteorological stations). A third possibility is to use TempLab to vary the temperature of sensor nodes using specific *test patterns*. For example, a user may not be interested in recreating a specific profile and needs instead only to verify whether a high temperature variation has an impact on the operation of a given protocol. In this case, TempLab can be fed with on-off patterns (e.g., a series of cold and warm periods) or jig-saw patterns that vary temperature with a specified frequency, allowing a quick debugging of protocols behaviour.



(a) Overview of our testbed infrastructure

(b) IR heating lamp on top of a sensor node

Figure 3.2: Overview of the TempLab testbed infrastructure at TU Graz, with infra-red heating lamps on top of each sensor node to control their on-board temperature [41].

3.1.3 Implementation

We have extended our local university testbed based on Maxfor MTM-CM5000MSP nodes with TempLab employing the following hardware and software components.

Hardware. We use Philips E27 IR 100W light bulbs that can be remotely dimmed using the Z-Wave wireless home automation standard. The latter operates on the 868 MHz ISM band, and hence does not interfere with the communications between the wireless sensor nodes (that use the 2.4 GHz ISM band). To vary the intensity of the light bulbs, we used Vesternet EVR_AD1422 Z-Wave Everspring wireless dimmers, which provide 100 dimming levels. Examples of LO nodes operating in the TU Graz facility are shown in Figure 3.2. Whereas LO nodes can only heat above room temperature, PE nodes have the capability of going below room temperature thanks to 4 cm-thick enclosures made of hard Polystyrene foam. In addition to the IR heating bulb, the enclosures contain an ATA-050-24 Peltier air-to-air assembly module by Custom Thermoelectric [64]. The latter is controlled through Vesternet EVR_AN1572 Z-Wave Everspring on-off wireless switches.

Software. We control the Z-Wave network with a C++ program that uses the Open Z-Wave stack to vary the intensity of dimmers and duty cycle the Peltier modules. Commands to the control network are sent through the Aeon Labs Series 2 USB Controller deployed within the testbed facility. Each node runs Contiki, and contains a low-priority process that periodically measures temperature using the on-board SHT11 sensor, and communicates the readings over the USB back-channel. The sampling frequency F_U, i.e., how often should the controller receive feedback about the on-board node temperature and update the intensity of the IR lamps, is selected by comparing the fastest temperature variation observed in the outdoor deployment shown in Figure 2.1 (1.98 °C/minute) and the accuracy of the on-board temperature sensors. In our case, the nodes carry SHT11 sensors that have an accuracy of 0.4°C and a variation of 1.98 °C/minute can be reached within 12 seconds.

The PI controller is implemented as a standalone multi-threaded C++ application executing on the testbed gateway that receives as input a file with two columns: the first one contains the time of the day, the second one describes the on-board temperature that the node should have at that time. The controller is agnostic to the type of trace (whether derived empirically or from a model): as long as the file adheres to the two column format, it will (try to) recreate such temperatures based on this information and the feedback signals from the motes. In case the user chooses to time-lapse the experiment, the controller skips rows accordingly, e.g., for a 2x speed, the controller omits every other line. Users can manually assign the available traces to the temperature-controlled nodes in the network, and if a non-implementable mapping is created, the controller will signal an error. The parameters of the controller P=2 and I=0.01 have been found empirically by testing the response of the system with extreme temperature variations and by choosing those values that achieve fast and self-stabilizing control as well as minimal overshooting.

3.1.4 Accuracy in replaying real-world traces

TempLab has been used to replay several traces from outdoor deployments collected during different times of the year, showing a high accuracy in reproducing a given temperature profile. TempLab's accuracy can be computed by analysing how close the instantiated temperature profile P_I follows the given profile to be reproduced P_G. The overall accuracy Q_n of the reproduced temperature profile at node n can be expressed as:

$$Q_n = \frac{1}{T}\int_0^T \left|\mathrm{P_I(t) - P_G(t)}\right| \mathrm{d}t \tag{3.1}$$

where T is the duration of the experiment. Besides the requirement to follow a temperature profile over time, it is also important to ensure that the rate of temperature changes is accurately reflected. At no point in time the instantiated temperature curve at a node n should deviate too much from the given temperature profile. The maximum deviation q_n can be expressed as:

$$q_n = \max_t |P_I(t) - P_G(t)| \quad \geq Q_n \tag{3.2}$$

The smaller the value of Q_n, the better the instantiation of the temperature profile, whereas the smaller q_n, the better the dynamics of the temperature change are reflected.

Figures 3.3 and 3.4 show the ability of TempLab to reproduce a trace collected in Wennerström et al.'s outdoor deployment in Uppsala, Sweden [220] during summer (August) using LO and PE nodes, respectively. In absence of time-lapsing, LO nodes and PE nodes can replay the desired temperature profiles with $Q_n = 0.18\,°\mathrm{C}$ and $0.12\,°\mathrm{C}$, and $q_n = 1.90\,°\mathrm{C}$ and $1.43\,°\mathrm{C}$, respectively, showing a remarkable accuracy. When the traces have been replayed using time-lapsing, LO nodes show evident limits due to the lack of cooling capabilities. Compared to the error of 0.18 °C at normal speed, the average error Q_n raises to 0.52 °C and 1.12 °C when the time is compressed by a factor of 3 and 5, respectively. Thanks to the Peltier cooling capabilities, the use of PE nodes reduces the error to $Q_n = 0.41\,°\mathrm{C}$ and 0.55 °C with a time-lapse factor of 3 and 5, respectively.

Figure 3.5 shows the ability of TempLab to reproduce a "colder" trace collected in Wennerström et al.'s outdoor deployment in Uppsala, Sweden [220] at the end of October,

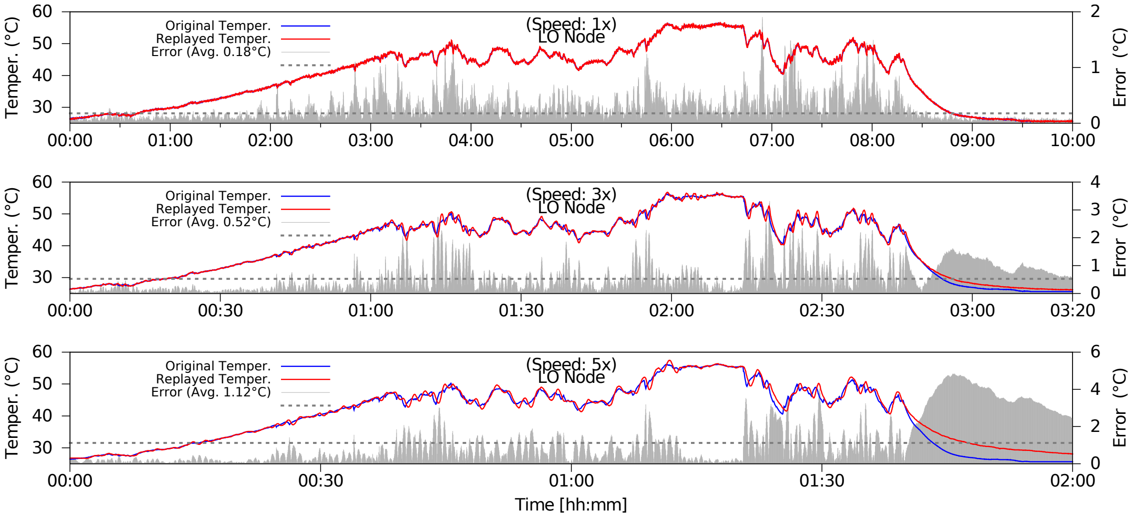


Figure 3.3: Accuracy of LO nodes in replaying a real-world trace captured during summer in Uppsala using TempLab with different time-lapsing factors [45].

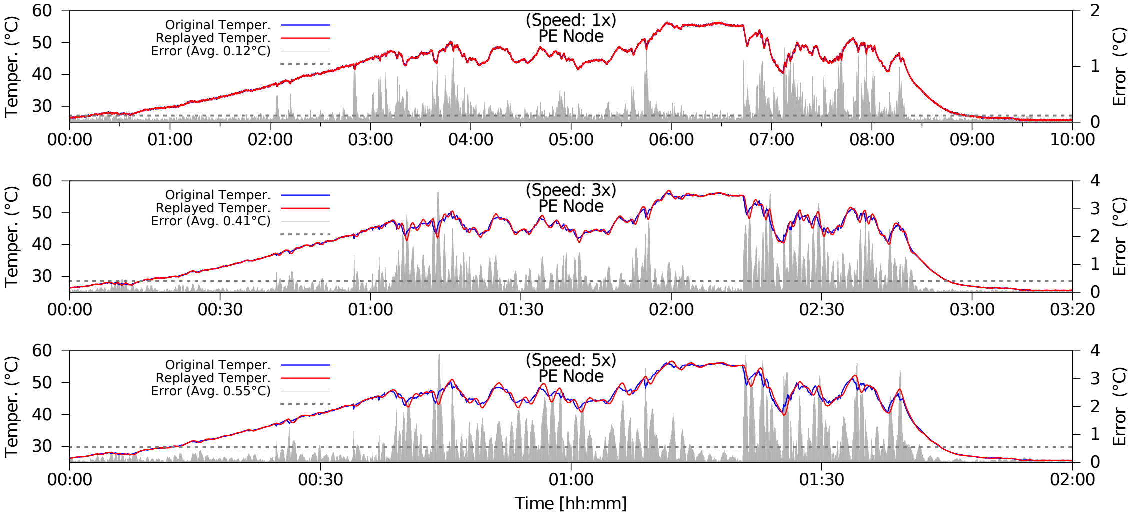


Figure 3.4: Accuracy of PE nodes in replaying a real-world trace captured during summer in Uppsala using TempLab with different time-lapsing factors [45].

when temperature approaches 0 °C. During winter time, the sun can quickly heat up the temperature in the package hosting the sensor nodes, and the trace indeed contains an on-board temperature variation from 45 °C during daytime to 0 °C in the evening. PE nodes replay these traces with an average error $Q_n = 0.14$ °C and 0.24 °C with a time-lapse factor of 3 and 5, respectively. The limit of PE nodes is reached when using a compression factor $T_F = 10$: a decrease in temperature of more than 20 °C (that in the real-world requires one hour and a half) cannot be replayed within 10 minutes only.

Figure 3.6 shows the replay of traces collected from the FIRE testbed facility in Santander, Spain. In particular, the traces are collected from nodes on the main road in

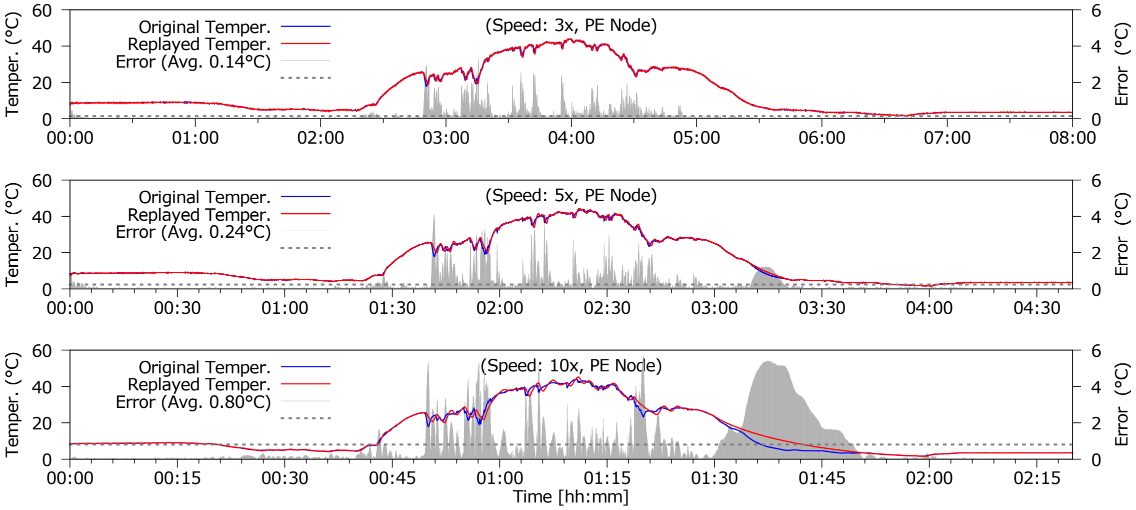


Figure 3.5: Accuracy of PE nodes in replaying a real-world trace captured during winter in Uppsala using TempLab with different time-lapsing factors [45].

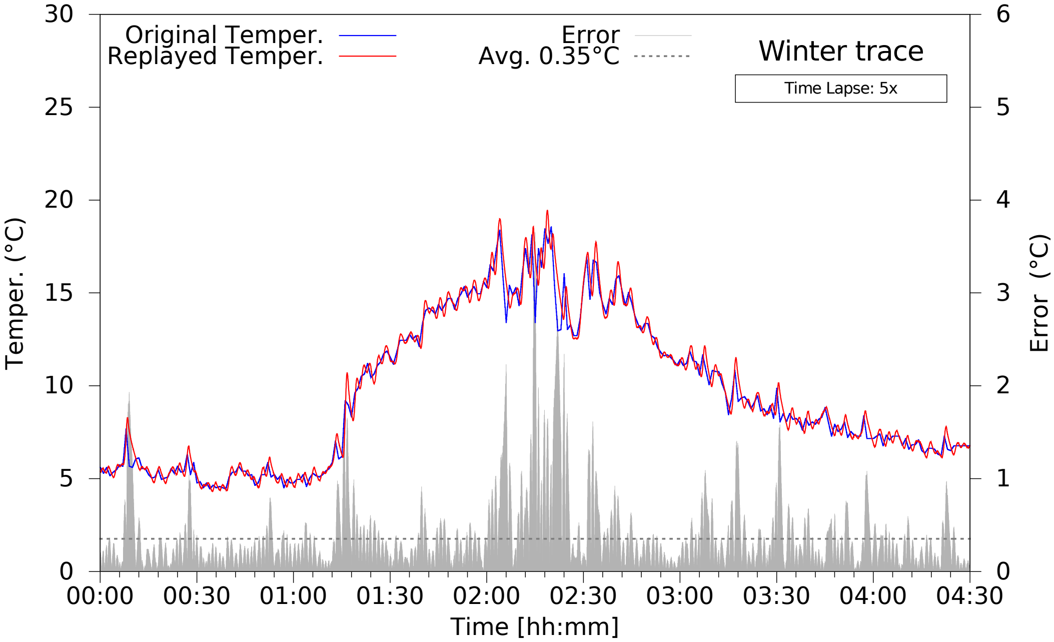


Figure 3.6: Accuracy of PE nodes in replaying a real-world trace captured during winter in Santander using TempLab with a time-lapse factor $T_F = 5$ [212].

Santander on the waterfront during winter and replayed on the testbed using PE nodes. The regeneration with a time-lapse factor $T_F = 3$ shows an average error $Q_n = 0.35\,°\mathrm{C}$, and a maximum error $q_n = 3.23\,°\mathrm{C}$. Using a compression factor $T_F = 20$ would increase the errors to $Q_n = 0.51\,°\mathrm{C}$, and $q_n = 5.96\,°\mathrm{C}$ [212].

3.2 Impact of Temperature on Wireless Link Quality

Several real-world outdoor deployments have experienced a degradation of the wireless link quality at high temperatures, as summarized in Section 2.2.4. Several works report a correlation between temperature fluctuations and PRR, whereas a few studies highlight an attenuation of the received signal strength at high temperatures.

To shed light on the impact of temperature on low-power wireless links, we tackle the problem in two consecutive steps. First, we carry out outdoor experiments and verify that higher temperatures actually reduce packet reception rate because of a decrease in the received power (Section 3.2.1). We then use TempLab to precisely characterize the dependency between link quality and temperature variations (Section 3.2.2), and derive a platform-independent model that characterizes the attenuation of signal strength on low-power radios (Section 3.2.3).

3.2.1 Packet reception

To analyse the impact of temperature on packet reception, we carry out several experiments at different locations to determine the minimum transmission power level necessary to ensure successful data transmission between two nodes[2].

We employ Tmote Sky nodes, and partition the deployed nodes into pairs consisting of a sending node and a receiving node running Contiki [72]. Given a pool of $N = 15$ packets with 12-byte payload, we define *the minimum power to reliably communicate* as the minimum power necessary to achieve 100% delivery, i.e., we expect exactly N received packets. Furthermore, we define *the minimum power to barely communicate* as the minimum power necessary to receive at least one of the N packets, without caring about the actual delivery rate.

Each sender transmits a train of N packets using Contiki's nullMAC, starting with the highest transmission power available. Each packet contains a sequence number and the information about the transmission power used by the sender. The receiving node uses the same transmission power as advertised in the message to reply to the sender. The receiver then sends an acknowledgement for every received packet, identified by its sequence number. If the sender receives at least one acknowledgement for the N packets sent, it will decrease the transmission power by one unit.

All the results obtained from our experiment runs show a significant increase in the minimum transmission power to both barely and reliably communicate at high temperatures. Figure 3.7, derived from a daily deployment in Lübeck, Germany during summer, shows the on-board temperature measured on two nodes not exposed to wind, placed approximately 7 meters away from each other. When the sun shines on the sensor nodes, the on-board temperature reaches up to 70 °C, with a variation of 55 °C compared to the average night temperature.

This high temperature variation causes an increase of the minimum transmission power to barely communicate from `PA_POWER`[3] 11 to 17, as well as an increase in the minimum transmission power to reliably communicate from `PA_POWER` 13 to 22. This corresponds to an increase in current consumption by 11.4% and 16.3%, respectively, and hints that,

[2] Full details about the experiments carried out can be found in Paper B included in this thesis [34].

[3] `PA_POWER` represents the register setting controlling the transmission power in the CC2420 radio. The register can be set with values between 0 (roughly -55 dBm) and 31 (0 dBm) [206].

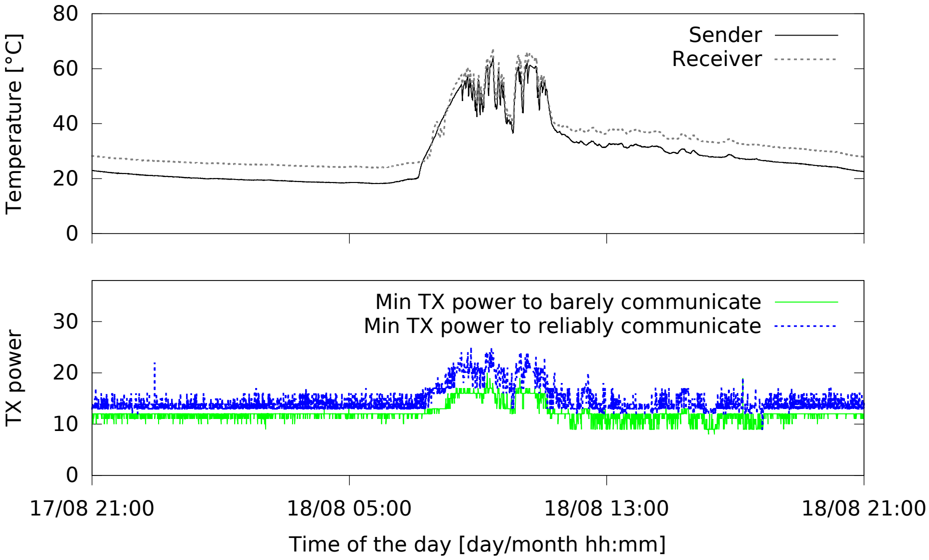


Figure 3.7: Minimum transmission power required by two sensor nodes to communicate. During daytime the sun shines directly on the motes, increasing significantly their on-board temperature. With respect to night-time operations, the sensor nodes require roughly 16% more power for a reliable transmission during the hottest time of the day [34].

in absence of power control strategies, signal strength attenuates at high temperature and can have a profound impact on packet reception.

We carry out a second experiment in Kista, Sweden during spring, in which we gradually vary the distance between the wireless sensor nodes from 50 cm to 20 m, and compare the minimum transmission power to barely communicate when temperature in both nodes is 18 °C and 38 °C respectively. Figure 3.8 confirms that temperature affects the minimum transmission power to successfully communicate regardless of the distance, hinting a specific radio issue. These results further indicate that reducing the transmission power during the coldest time of the day or the year may lead to significant energy savings.

3.2.2 Signal strength attenuation

To get a deeper understanding of the effects observed in Section 3.2.1, we use TempLab to precisely characterize the dependency between link quality and temperature variations[4]. In particular, our first step is to isolate the effects of temperature on transmitting and receiving nodes and to systematically study the impact of temperature on different hardware platforms, namely Maxfor MTM-CM5000MSP and Zolertia Z1 nodes employing the CC2420 radio [206], and Arago Systems WisMotes employing the CC2520 transceiver [205].

We partition the sensor nodes in our TempLab testbed facility into pairs and form bidirectional links operating on different physical channels to avoid internal interference. All sensor nodes run the same Contiki software: each sensor node continuously measures the ambient temperature and relative humidity using the on-board SHT11 or SHT71

[4] Full details about the experiments carried out can be found in Paper E included in this thesis [44].

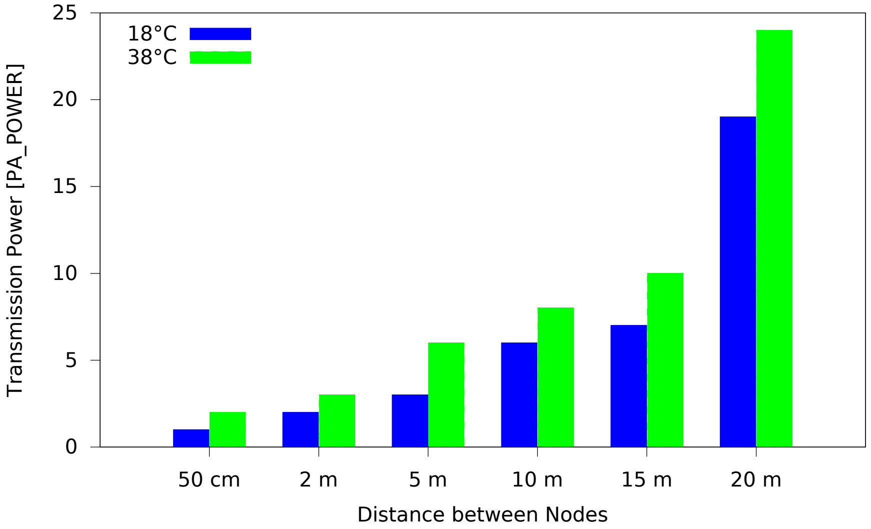


Figure 3.8: Minimum transmission power required to barely communicate between two sensor nodes at different distances. Regardless of the distance between the motes, higher on-board temperatures require an higher transmission power to maintain communication [34].

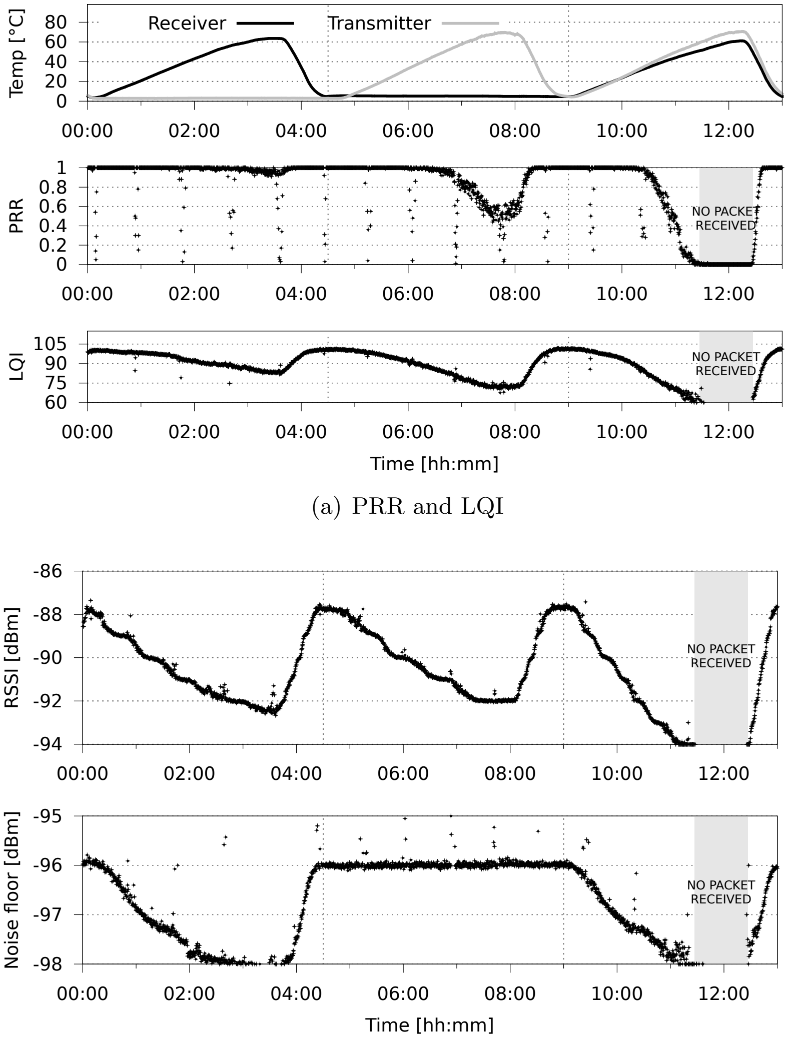
digital sensors, and periodically sends packets to its intended receiver at a rate of 128 packets per second using different transmission power levels. Statistics about the received packets are logged using the USB back-channel and are available remotely.

In a first experiment using Maxfor nodes, every link in the testbed is exposed to three heat cycles generated using TempLab's open-loop controller. First, each individual node, i.e., first the receiver and then the transmitter, is heated from 0 up to 65 °C. Afterwards, both nodes are heated in the same temperature range at the same time.

Figure 3.9(a) illustrates the impact of temperature on PRR and LQI on a particular link. The evolution of temperature at the transmitter and at the receiver over the 13-hours experiment is shown in the top figure. In correspondence to each increase of temperature, PRR and LQI decrease significantly, with the highest impact occurring when both nodes are heated. With both nodes heated, indeed, no packet was received and the connectivity between the two nodes was impaired until the temperature started to decrease.

Figure 3.9(a) also shows that the packet loss rate is more pronounced when the transmitter is heated compared to the case in which only the receiver is heated. We have observed this behaviour for the majority of links in our testbed.

Figure 3.9(b) illustrates the impact of temperature on RSSI (top figure) and noise floor (bottom figure). The former represents the signal strength of each packet received from the transmitter node; the latter refers to the signal strength measured in absence of any packet transmission. The RSSI decreases in a similar way when transmitter and receiver are heated separately, whereas the decrease is more pronounced if both transmitter and receiver are heated at the same time. This proves that temperature decreases both the transmitted and received power [23], whereas the noise floor only decreases when the receiver node is heated, with an absolute variation smaller than the one of RSSI.



(a) PRR and LQI

(b) RSSI and Noise floor

Figure 3.9: Impact of temperature on the quality of links. Using TempLab, we heat transmitter and receiver nodes separately first, and then both of them at the same time. When temperature increases, PRR, LQI, and RSSI decrease significantly, with the highest impact occurring when both nodes are heated at the same time [44].

Furthermore, the decrease in RSSI does not seem to depend on how quickly temperature changes: in our setup, the heat cycles are characterized by a slow increase in temperature followed by a quicker cooling phase, as can be seen in Figure 3.9(a). This allows us to observe that both RSSI and noise floor are not affected by how quickly temperature varies, and that the impact of temperature can be modelled using the absolute temperature value at the transmitter and receiver nodes.

We repeat the same experiment using different hardware platforms and specifically study the decrease in RSSI. Our experimental results show that the RSSI decreases in an approximately linear fashion with temperature, and that the relationship varies depending

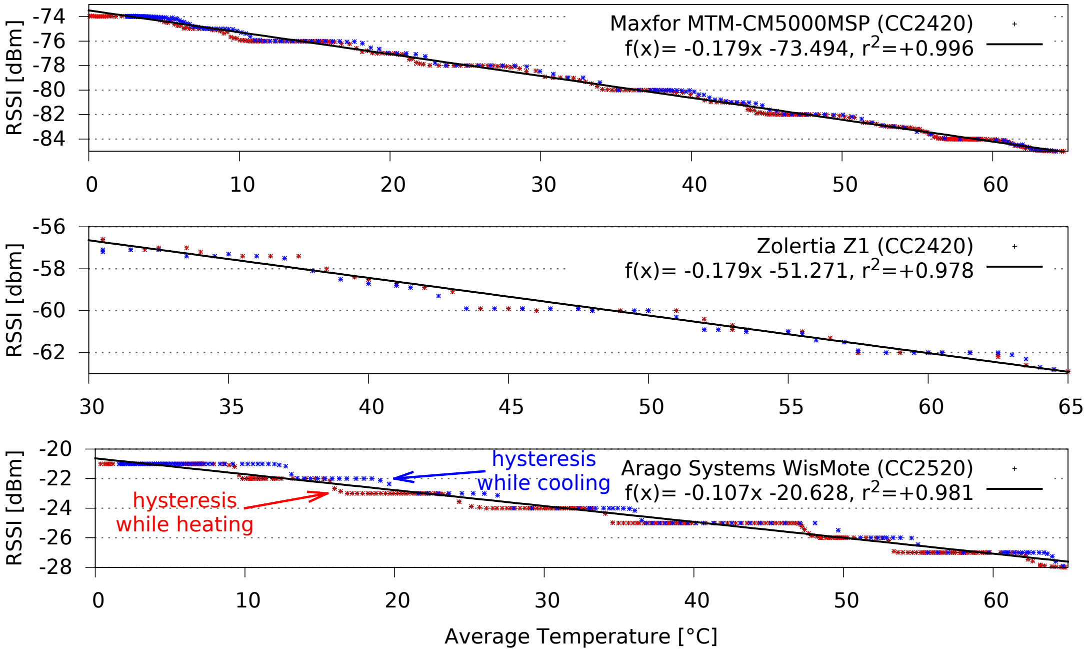


Figure 3.10: The relationship between RSSI and temperature can be approximated as a linear function with different parameters depending on the hardware platform employed [44].

on the radio chip employed. Figure 3.10 shows the relationship between RSSI and temperature obtained on different platforms when heating both nodes at the same time. The hardware platforms employing the same CC2420 radio exhibit approximately the same slope, whereas the WisMotes show a slightly lower attenuation with a more visible *hysteresis*. We apply linear regression to derive the equations and compute how accurately they represent our experimental datasets by calculating the coefficient of determination (r^2). Figure 3.10 shows that linear regression allows us to accurately capture the relationship between RSSI and temperature: the coefficient of determination r^2 varies between 0.978 and 0.996.

Figure 3.11 shows the relationship between RSSI and temperature obtained using Maxfor nodes when transmitter and receiver nodes are heated individually and when both nodes are heated at the same time (top and bottom figures refer to the same link, but are obtained using a different transmission power). The relationship between RSSI and temperature is approximately the same, but it exhibits a slightly steeper decrease when the receiver is heated. Although a comparison between curves is difficult due to the Automatic Gain Control (AGC) operations (i.e., depending on whether we capture the transition between two discrete steps, we may obtain slightly different slopes), by averaging the data from all our experiments we can obtain the slope of the linear function. The equations shown in Figure 3.11 are derived as explained previously and the coefficient of determination r^2 shows how accurately they match the experimental datasets.

A decrease in SNR leads to a lower link quality and a shorter radio link, which in turn may lead to lower throughput, higher delay or even network partitioning. We therefore

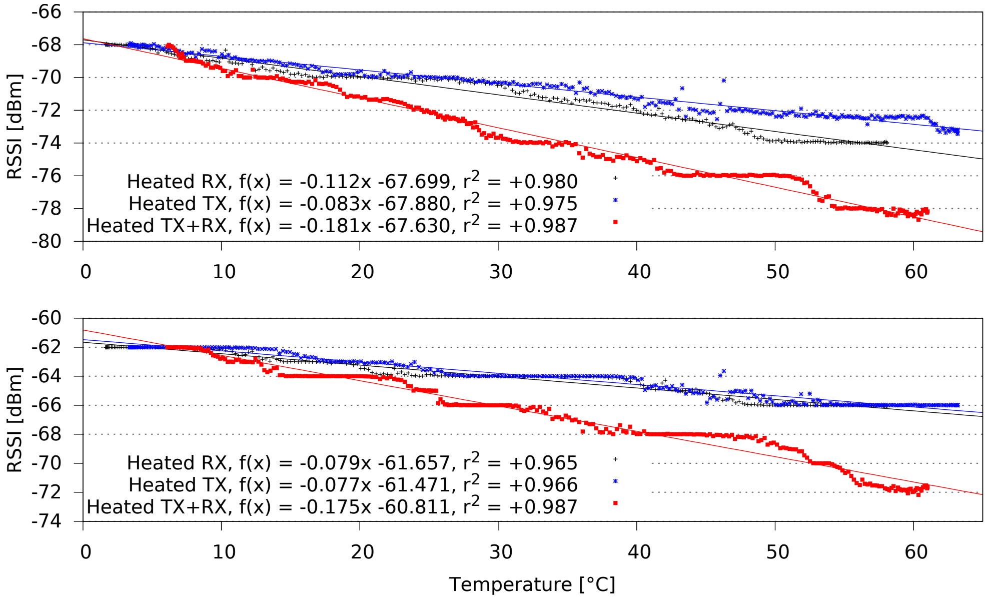


Figure 3.11: Relationship between RSSI and temperature when transmitter (blue) and receiver (black) nodes are heated individually, and when both nodes (red) are heated at the same time [44].

ultimately analyse how the SNR of a link is affected by temperature. Figure 3.12 illustrates how noise floor, RSSI, and SNR vary on a given link when transmitter and receiver nodes are heated individually and at the same time. Since the noise floor decreases only when the receiver is heated, an increase in temperature on the transmitter has a higher impact on the SNR compared to an increase in temperature at the receiver (the slope of the black curve is rather similar in the bottom figure, but significantly steeper in the top figure). This result is consistent with the different packet reception rates observed in Figure 3.9(a) when nodes were heated individually. During the first heating cycle, only the receiver was heated and the PRR was only affected slightly, whereas during the second heated cycle, when only the transmitter node was heated, PRR dropped to 40%.

3.2.3 Platform-independent analytical model

We now derive an analytical model that captures the decrease in SNR as a function of temperature for a generic platform.

Denoting PL as the path loss between a transmitter-receiver pair, P_t as the transmission power, P_r as the received power, and P_n as the noise floor at the receiver, the SNR is known to be:

$$\begin{aligned} SNR(dB) &= P_t - PL - P_n \\ &= (P_t - P_n) - (P_t - P_r) \\ &= P_r - P_n \end{aligned} \tag{3.3}$$

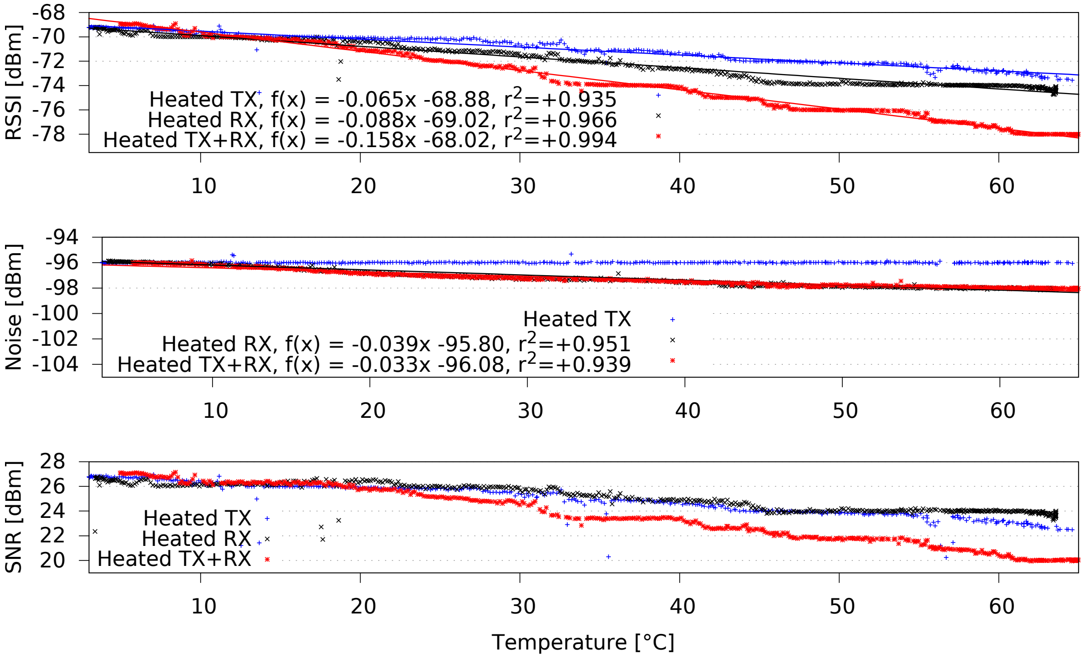


Figure 3.12: Relationship between RSSI, noise floor, SNR, and temperature when transmitter (blue) and receiver (black) nodes are heated individually, and when both nodes (red) are heated at the same time [44].

The empirical measurements presented in Section 3.2.2 have shown that an increasing temperature has three main effects on the signal strength of radio transmissions; it (i) decreases the transmitted power, (ii) decreases the received power, and (iii) decreases the noise floor. Denoting α, β, γ as constants with units dB/K, T_t and T_r as the reference temperature in Kelvin of transmitter and receiver, ΔT_t and ΔT_r as the difference in Kelvin with respect to T_t and T_r, the effect of temperature on SNR can be modelled as:

$$\begin{aligned} SNR &= (P_t - \alpha\Delta T_t) - (PL + \beta\Delta T_r) \\ &\quad -(P_n - \gamma\Delta T_r + 10\log_{10}(1 + \tfrac{\Delta T_r}{T_r})) \\ &= P_t - PL - P_n - \alpha\Delta T_t \\ &\quad -(\beta - \gamma)\Delta T_r - 10\log_{10}(1 + \tfrac{\Delta T_r}{T_r}) \end{aligned} \tag{3.4}$$

The proportional relation between ΔT and the constants α (effect on transmitted power), β (effect on received power) and γ (effect on noise floor) is based on the empirical observations made in the previous sections. The term $10\log_{10}(1 + \frac{\Delta T_r}{T_r})$ is derived analytically from the Johnson-Nyquist thermal noise equation.

There are two important trends to highlight in this model. First, changes in temperature have a higher impact on the transmitted and received powers (linear relation of α and β), than on the thermal noise (logarithmic relation). Second, to some extent it is counter-intuitive that a higher temperature decreases the noise floor (negative sign of γ). This effect was also observed by Bannister, and he hypothesizes that it is due to the losses in the signal amplifier [22]. That is, a higher temperature not only reduces the gain of the signal but also the gain of the noise. Hence, the received signal strength is lower for both.

The accuracy of our model depends on identifying the right values for α, β, and γ. The latter are platform-dependent and correspond to the slopes of the linear trends observed in our empirical results. To parametrize the model, it is hence sufficient to carry out a systematic and fine-grained evaluation of the platform of interest using TempLab. For example, a network manager willing to deploy a network using the Maxfor MTM-CM5000MSP platform, can use TempLab to compute the slopes obtained in Figure 3.12.

3.3 Impact of Temperature on Protocol Performance

Communication protocols often rely on signal strength readings and link quality estimation metrics. As the results presented in Section 3.2 hint that temperature variations are likely to harm their operation, we analyse in this section the impact of temperature on state-of-the-art routing and MAC protocols.

3.3.1 Routing layer

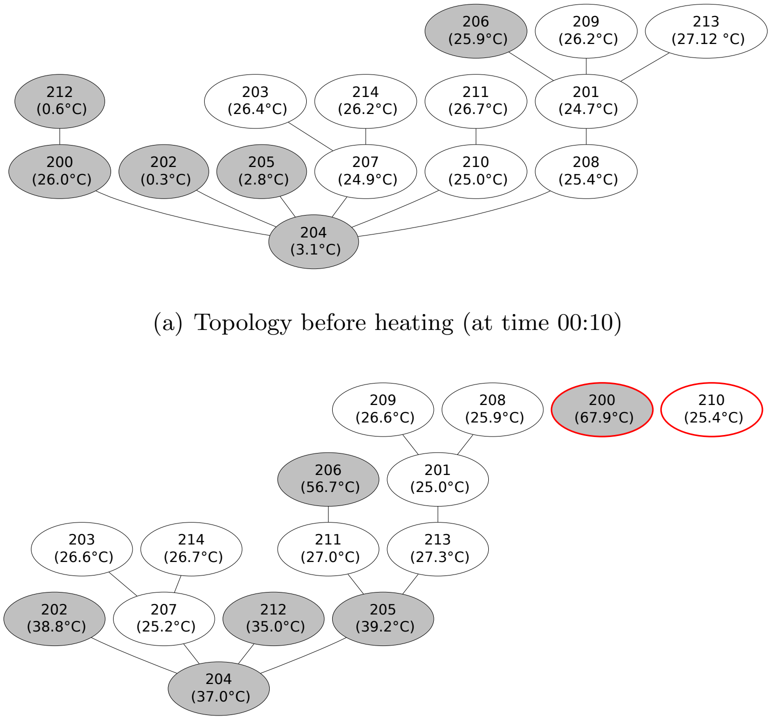
Bannister et al. [23] have hypothesized that network connectivity can significantly degrade at high temperatures, as a result of signal strength attenuation.

We verify the problem experimentally by studying how temperature fluctuations affect the behaviour of the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [225]. We program fifteen Maxfor MTM-CM5000MSP nodes in our local testbed at TU Graz with a basic Contiki application that uses ContikiRPL [211]: each node sends a message to the root node every minute and logs the transmitted and received packets, as well as the on-board temperature and the expected number of transmissions (ETX) of active links[5]. We use TempLab to evaluate the impact that daily fluctuations of temperature can have on the RPL topology with a test pattern that gradually increases the temperature of the designated root node and of one third of the other testbed nodes.

Figure 3.13(a) shows a snapshot of the RPL topology at the beginning of the experiment, when nodes are kept at low temperature: all nodes are connected to the sink within a maximum of three hops. Figure 3.14 (top) shows how temperature evolves during the experiment, whereas Figure 3.13(b) illustrates the RPL topology after temperature has increased: temperature-controlled nodes are shaded in gray.

Network partition and increase in network diameter. The increase in temperature led to drastic changes in the topology of the network, including a network partition and an increase in network diameter. Nodes 200 and 210 had a direct link to the root node when temperature was low (Figure 3.13(a)), but these links are isolated from the network once temperature has increased. Indeed, ContikiRPL attempts to construct a tree by minimizing the ETX sum along the paths to the root. However, ETX changes abruptly with fast temperature changes, especially when packets are exchanged sporadically. In our experiments, we can indeed observe a sudden increase of ETX in links 200 → 204 and 210 → 204 (Figure 3.14), which will lead to a sudden network partition and a significant energy waste until a new tree is computed.

[5] Full details about the experiments can be found in Paper F included in this thesis [45].



(a) Topology before heating (at time 00:10)

(b) Topology after heating (at time 01:00)

Figure 3.13: An increase in temperature can lead to drastic changes in the RPL topology, including a network partition and an increase in network diameter [45]. Temperature-controlled nodes are shaded in gray.

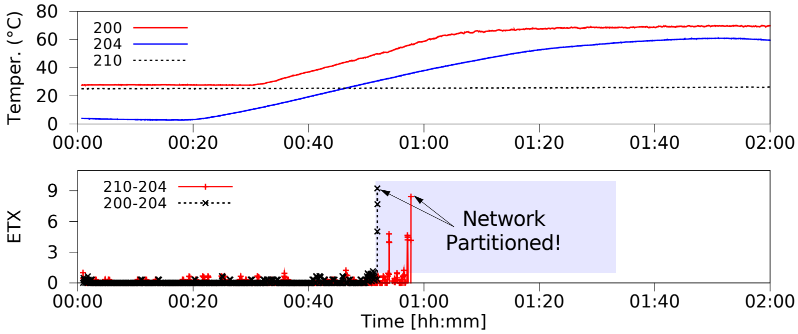
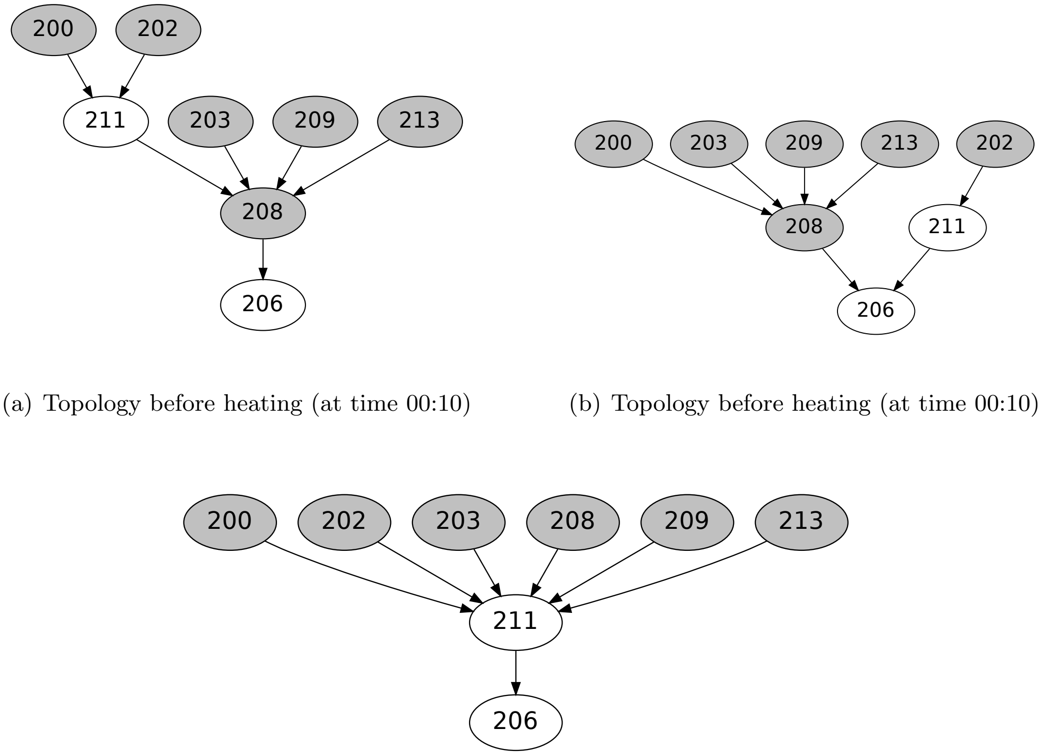


Figure 3.14: Sudden rise of ETX while temperature increases [45].

Bottlenecks in the topology. In the previous experiments, only a fraction of the nodes were exposed to a gradual temperature variation: this was sufficient to cause disconnections in the network and to increase the network diameter. We have repeated the same experiment on a reduced scale with all nodes being heated using TempLab except the sink node, and one of its neighbours (emulating the case in which one node was shadowed by vegetation, and its on-board temperature did not raise significantly compared to the rest of the other nodes in the network).



(a) Topology before heating (at time 00:10)

(b) Topology before heating (at time 00:10)

(c) Topology after heating (at time 01:00)

Figure 3.15: An increase in temperature can lead to drastic changes in the network topology, and nodes with minimal on-board temperature variations may be selected as forwarding nodes by several nodes and become overloaded. Temperature-controlled nodes are shaded in gray.

Figures 3.15(a) and 3.15(b) show the network topology at the beginning of the experiment when the on-board temperature of the nodes has not been altered by TempLab. Temperature-controlled nodes are shaded in gray. We show two different topologies to highlight the stochastic nature of the topology formation on RPL, which depends on the *trickle timers* used on nodes to announce Directed Acyclic Graph (DAG) information object messages: repeatability on a topology level cannot be ensured in these settings.

Figure 3.15(c) shows the topology of the network after temperature has increased (this topology remained consistent across different experiment regardless of the initial topology). Node 211 is used as primary forwarder to the sink from all other nodes in the network. When increasing the network load, we experienced a significant performance deterioration, as node 211 was overloaded by messages originating from its children.

These results emphasize the need for techniques that forward the on-board temperature information of neighbours to the routing layer, so that the most stable tree can be computed before drastic temperature changes occur [117], for example using the first-order model proposed in Section 3.2.3.

3.3.2 MAC layer

In Carrier Sense Multiple Access (CSMA) protocols, the correct operation of Clear Channel Assessment (CCA) is fundamental to reduce the number of wasteful transmissions and to preserve the limited energy budget of the nodes in the network. The typical task of CCA is to avoid collisions, i.e., to determine whether another device is already transmitting on the same frequency channel. If there are ongoing transmissions, CSMA protocols defer transmissions using different back-off strategies [43]; otherwise packets are immediately transmitted. CCA is also used in low-power duty-cycled MAC protocols to trigger wake-ups, i.e., to determine if a node should remain awake to receive a packet or whether it should return to sleep mode [164]. Towards this goal, low-power MAC protocols typically perform an inexpensive CCA check and keep the transceiver on if some ongoing activity is detected on the channel [49, 71, 164].

CCA implementations are typically based on energy detection, i.e., on the measurement of the received signal strength and on its comparison with a given threshold. When performing energy detection using a fixed CCA threshold (most state-of-the-art protocols actually employ a fixed threshold), it is neglected that received signal strength readings are affected by temperature, and this leads to a number of problems. First, the transmitter can erroneously measure weaker noise in the environment as a result of the increased temperature, and generate wasteful transmissions. Second, a receiver node may not receive a signal sufficiently strong to cause a wake-up of the radio, and constantly remain in low-power mode at high temperatures, causing the disruption of the link.

Inefficient Collision Avoidance When a protocol employs a fixed CCA threshold T_{CCA} to determine whether another device is already transmitting, it essentially neglects that the received signal strength depends on the temperature. We now show experimentally that this can lead to an increase in false negatives when a transmitter is assessing the presence of a busy medium at high temperatures[6].

We carry out experiments consisting of several transmitter-receiver pairs running a basic Contiki application, in which the transmitter node periodically sends packets to its intended receiver and collects statistics such as the energy expenditure and the RF noise in the radio channel [73]. The latter is computed as the maximum of 20 consecutive RSSI readings after a packet transmission. In a first experiment in an environment rich of Wi-Fi interference, we use Contiki's nullMAC and nullRDC to avoid protocol-specific implementations and employ the CC2420's default CCA threshold (-77 dBm). We use TempLab to vary the on-board temperature of the nodes between 25 and 75°C. Except from temperature, there is no significant change in the environment surrounding the nodes.

Figure 3.16 shows the RF noise captured using RSSI readings by a node in our testbed. The noise has a visible correlation with the on-board temperature of the node, and follows the attenuation described in Section 3.2.2. We can observe that at around 40°C, there is an intersection between the measured signal strength and the selected CCA threshold T_{CCA}. For temperatures lower than 40°C the measured RSSI is above T_{CCA} (and hence transmissions would be deferred); for temperatures higher than 40°C, instead, the RSSI is below T_{CCA} (and packets would be immediately sent). In other words, the MAC protocol erroneously deduces from RSSI readings obtained above 40°C that the channel is free from

[6] Full details about the experiments can be found in Paper G included in this thesis [41].

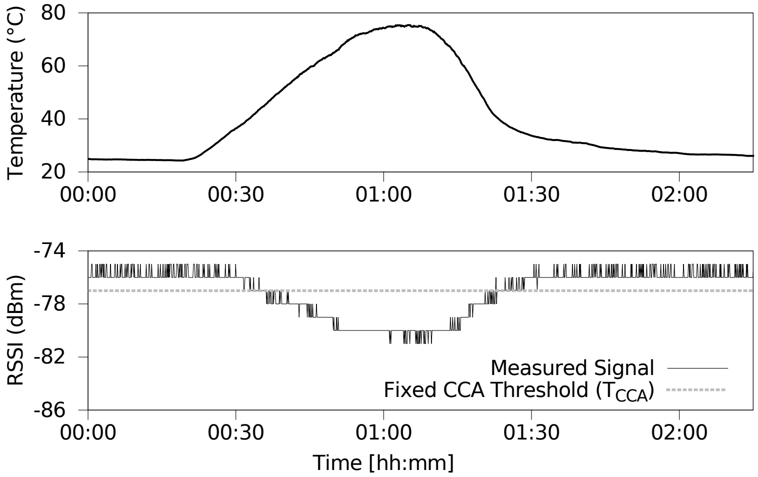


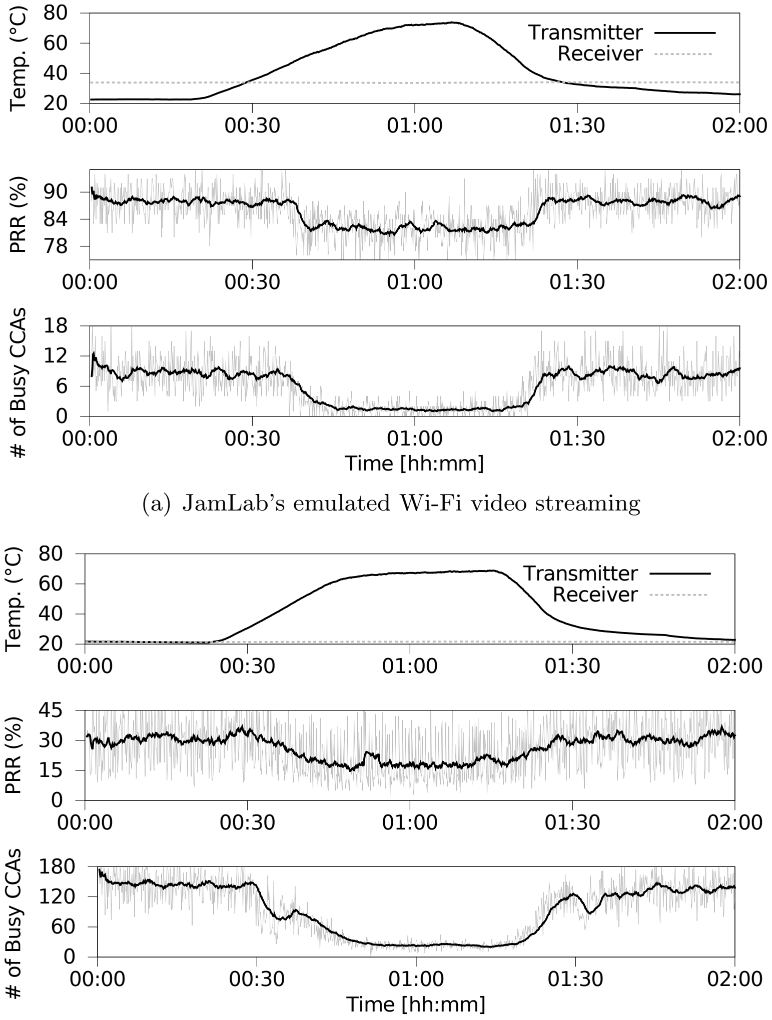
Figure 3.16: The signal strength attenuation at high temperatures can cause an intersection with the clear channel asseessment threshold T_{CCA}, causing several issues [41].

harmful interference. In reality, the interference in the environment is not weakened by temperature (the RSSI attenuation is only an artefact of the radio), and can still destroy transmitted packets. These erroneous CCAs at high temperature may hence lead to an increase in the number of wasteful transmissions destroyed or corrupted by interference.

Figure 3.17 shows the impact of erroneous CCAs in the presence of different interference patterns. We use JamLab [42] (see Chapter 4.2) to produce repeatable interference in our testbed on different channels. We emulate on one channel the interference caused by a computer streaming videos from a Wi-Fi access point, and on another channel the interference caused by an active microwave oven. We also let a computer transfer large files from a nearby Wi-Fi access point using a channel that is not affected by JamLab. We then analyse how this affects the PRR on the transmitter-receiver pairs in our testbed that experienced an intersection between measured noise and T_{CCA} at different temperatures as in Figure 3.16.

The PRR decreases in all scenarios as soon as the on-board temperature of sensor nodes increases. In the presence of Wi-Fi video streaming, the PRR of the link decreases from 88% to 81% (Figure 3.17(a)), whereas in the presence of an active microwave oven the PRR decreases from 70% to 45% [41]. Similarly, also the PRR in the presence of a Wi-Fi file transfer decreases from 30% to 18% at high temperatures (Figure 3.17(b)). We can also notice that the decrease in PRR is correlated with a decrease in the number of CCAs identifying a busy channel, i.e., with a decrease in the number of CCAs that do not identify potential collisions at high temperatures. These results prove that the intersection between the RSSI curve and the CCA threshold shown in Figure 3.16 results in erroneous clear channel assessments leading to a decreased PRR at high temperatures.

Unsuccessful wake-up of nodes High temperatures can also affect the correctness of clear channel assessment when waking-up the transceiver from sleep mode. State-of-the-art MAC protocols often duty cycle the radio to reduce energy consumption, and employ clear channel assessment to wake-up the transceiver from sleep mode. Typically, a periodic



(a) JamLab's emulated Wi-Fi video streaming

(b) File transfer between two Wi-Fi devices

Figure 3.17: Temperature affects the efficiency of collision avoidance in CSMA protocols. Our experiments in different interference scenarios show that when the received signal strength weakens to values below T_{CCA} at high temperatures, the PRR decreases, as well as the number of CCAs identifying a busy channel [41].

CCA check is performed: if the channel is busy, the transceiver is kept awake in order to receive the incoming packet, otherwise the radio returns to sleep mode.

Imagine a sender A and a receiver B exchanging packets using a duty-cycled MAC protocol in which A sends short strobes before the actual packet (or repeatedly sends the same packet). If B receives the strobes from node A with a signal strength that is higher than T_{CCA}, it keeps its radio on and receives the payload message from A. If temperature increases, the received signal strength at node B may intersect T_{CCA} as shown in Figure 3.16. When this happens, the transmissions from A are received with a signal strength lower than T_{CCA}, and B does not wake up to receive A's packets anymore, essentially disrupting the link. In the case shown in Figure 3.16, the link would be

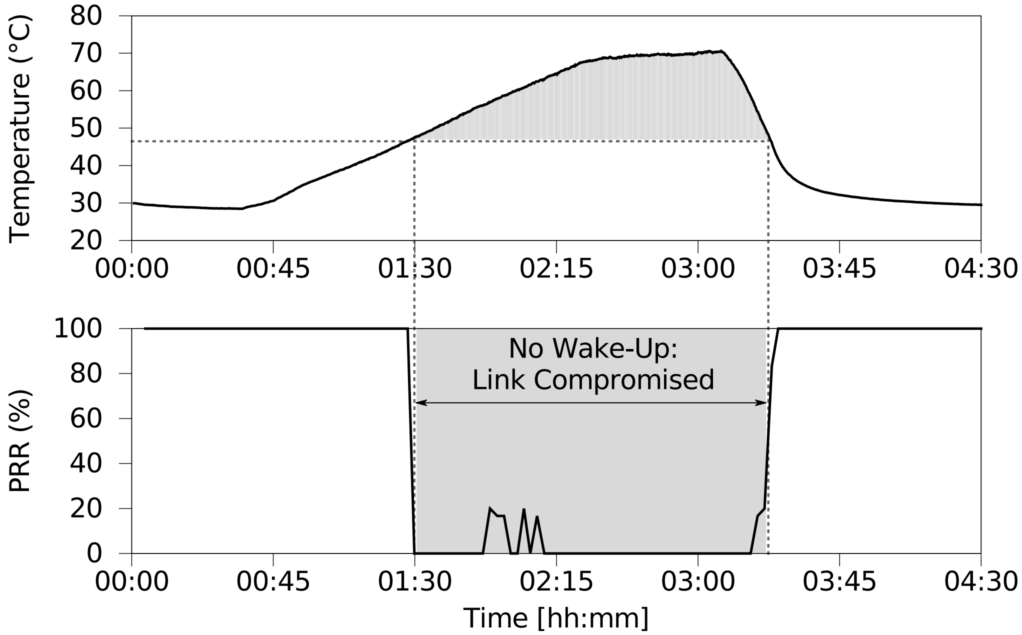


Figure 3.18: Temperature can affect the wake-up mechanism in duty-cycled MAC protocols. When the strength of the received signal from a transmitter weakens at high temperatures and intersect the CCA threshold as shown in Figure 3.16, the receiver does not wake up anymore, disrupting the link's connectivity [41].

disrupted for temperatures higher than 40°C, because node B would not wake up when the strength of the received signal from A decreases below T_{CCA}.

We show experimental evidence of this problem by letting several transmitter-receiver pairs of nodes communicate using ContikiMAC, Contiki's default MAC protocol in which nodes sleep most of the time and periodically wake up to check for radio activity. In ContikiMAC, the transmitter sends repeatedly the same packet until a link layer ACK is received, whereas the receiver keeps its radio on as soon as a packet transmission is detected by means of a single CCA check [71]. Packets are exchanged every 20 seconds, and ACKs are sent using CC2420's hardware support. As in the previous experiment, we use TempLab to warm-up and cool-down the on-board temperature of the nodes, emulating the daily fluctuations that can be found in real-world deployments.

Figure 3.18 shows an example of link disruption caused by a receiver not waking up at high temperatures. We can notice that what was a perfect link until approximately 47°C, suddenly does not receive any packet at higher temperatures. Only once temperature decreases below 47°C, the link is restored and the node correctly receives the packets sent from the transmitter. This behaviour can significantly harm network performance, as links may disappear during the hottest times of the day, leading to high latencies, drastic topology changes, or in case no alternative paths for communication can be found, to a complete disconnection of some nodes from the network.

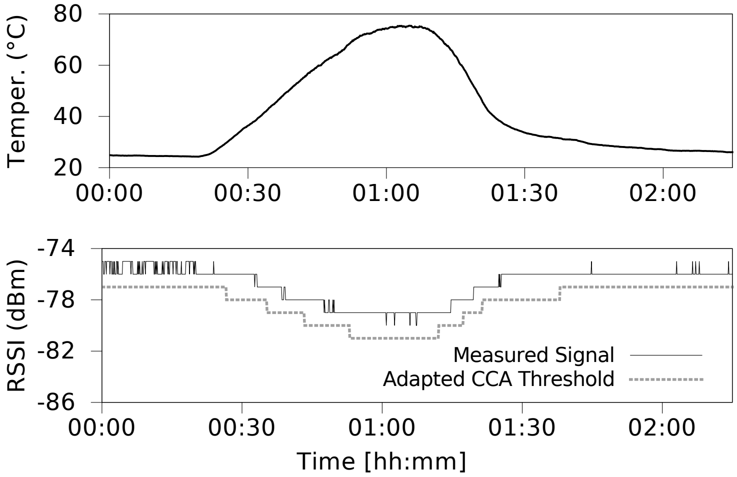


Figure 3.19: Dynamic adaptation of the CCA threshold based on the temperature measured locally on the node: T_{CCA} follows the attenuation of the signal, avoiding an intersection with the RSSI curve (in contrast with Figure 3.16) [41].

3.4 Mitigating the Adverse Effects of Temperature on Protocol Performance

Exploiting the model capturing the relationship between signal strength attenuation and temperature presented in Section 3.2.3, we now propose an adaptive technique to mitigate the inefficiency of CSMA protocols at high temperatures[7].

3.4.1 Predicting the attenuation of signal strength

Imagine a sender A and a receiver B exchanging packets. Following the observations made in Section 3.2.3, if the on-board temperature of B varies by ΔT_B degrees w.r.t. to an initial temperature τ, the signal will suffer an attenuation on the receiver side by $R = \beta\Delta T_B$, with ΔT_B being the difference between B's current temperature T_{now} and τ. Similarly, if the on-board temperature of A varies, its signals will be transmitted with an attenuation on the transmitter side of $T = \alpha\Delta T_A$, and B will receive a signal that is T dBm weaker. In case the temperatures of both A and B vary, the overall attenuation of the received signal strength on B is given by $R + T$. If temperature has decreased, $\Delta T = (T_{now} - \tau)$ is negative, and R and T do not represent an attenuation, but instead a strengthening of the signal.

α and β are specific to the employed radio and can be characterized by computing the variation of signal strength on a large temperature range and by deriving the slope of the RSSI curves of transmitter and receiver for a given platform as in Section 3.2.3. In the case of the Maxfor MTM-CM5000MSP nodes employed in our experiments we derive $\alpha = \beta =$ -0.08 dB/°C. We further model the attenuation of the noise floor as $N = \gamma\Delta_T$ (which is typically smaller than R and T) and derive $\gamma =$ -0.05 dB/°C.

[7] Full details can be found in Paper G included in this thesis [41].

3.4.2 Adapting the CCA threshold at runtime

Exploiting the above model, we compute the attenuation of signal strength caused by temperature variations, and adapt the CCA threshold at runtime. Each node needs to compute if the temperature varied significantly enough to cause an attenuation of the signal strength w.r.t. an initial threshold $T^{'}_{CCA}$.

The initial CCA threshold is typically fixed (e.g., -77 dBm by default in the CC2420 radio). However, as nodes are typically uncalibrated and have radio irregularities, a good practice would be to select $T^{'}_{CCA} = n^{'}_{f} + K$, with $n^{'}_{f}$ being the noise floor of the node, and K a constant defined at compile time.

If this is the case, $T^{'}_{CCA}$ and $n^{'}_{f}$ are computed during start-up, while a node experiences an on-board temperature τ. If $T^{'}_{CCA}$ is fixed, we assume $\tau = 25°\text{C}$. Please note that high values of K reduce the number of activities in the channel that can trigger a wake-up of a node (minimizing energy consumption), but also reduce the number of links in the network (fewer neighbours can wake-up a node with a signal strength higher than $T^{'}_{CCA}$).

Whenever temperature varies significantly, we compute the updated threshold as $T_{CCA} = T^{'}_{CCA} + T + R$, with T and R being computed using the difference between the current temperature and τ. All that is needed to adapt the threshold is hence an up-to-date information about the current on-board temperature of the nodes and the initial temperature τ stored in a 2-byte variable. We apply to the computation of T_{CCA} a lower bound $n_f + C$ (with $n_f = n^{'}_{f} + N$) that avoids the selection of CCA thresholds that are too close to the noise floor (this would cause the radio to continuously wake-up).

Figure 3.19 shows the adaptation of the CCA threshold based on the algorithm detailed previously. If we compare the results with the ones shown in Figure 3.16, we can notice that the CCA threshold follows the same attenuation as the received signal, avoiding an intersection between the RSSI curve and T_{CCA}. Hence, the model proposed in Section 3.2.3 is sufficiently accurate to dynamically adapt T_{CCA} to local temperature changes.

Local adaptation. By periodically sampling the on-board temperature, a node can compare its current temperature with τ and compute Δ_T, immediately deriving N and R. If a node would adapt its CCA threshold based on this information (i.e., using $T = 0$), the inefficient collision avoidance problem at high temperatures would be solved, as well as the wake-up problem in case the temperature of the transmitter does not vary significantly.

Such a computationally inexpensive *local adaptation* of the CCA threshold can be easily applied to all duty-cycling protocols and can significantly mitigate the adverse effect of temperature variations on communication. However, problems are not mitigated completely: if a node receives packets sent from a node experiencing temperature fluctuations, it would need to know the temperature of the transmitter to derive T and completely mitigate the unsuccessful wake-up problem. This is non-trivial, as a receiver does not necessarily know the identity of the sender by the point in time in which it performs a CCA, and as it may actually be recipient of packets sent by different nodes.

The information about the transmitter's temperature can be conveyed by the network layer, which could augment the table of neighbour addresses and attributes with the latest on-board temperature of each neighbour. If a modification of the network layer is suitable for the considered system, we propose a cross-layer approach to derive T.

Cross-layer adaptation. To make more informed decisions, we use existing routing beacons to piggyback temperature information efficiently and to compute the maximum temperature change across all neighbours. We implement this by using RPL, and disseminate the temperature information by piggybacking it on the routing beacons. RPL sends these beacons to the neighbour nodes with quickly increasing time intervals, as regulated by the Trickle algorithm [128]. Within the Destination-Oriented DAG[8] Information Object (DIO), there is room to embed a routing metric container object, which holds different parameters and constraints that are used to take routing decisions. Beside the metric container specified in the standard, it is possible to use implementation-defined metric containers. We make each node report its current and maximum temperature through such a metric container. Once a node receives this information in an incoming routing beacon, it stores it as an attribute in Contiki's neighbour table, from whence it can be retrieved by the CCA adaptation module to calculate the maximum temperature change in the neighbourhood.

3.4.3 Evaluation

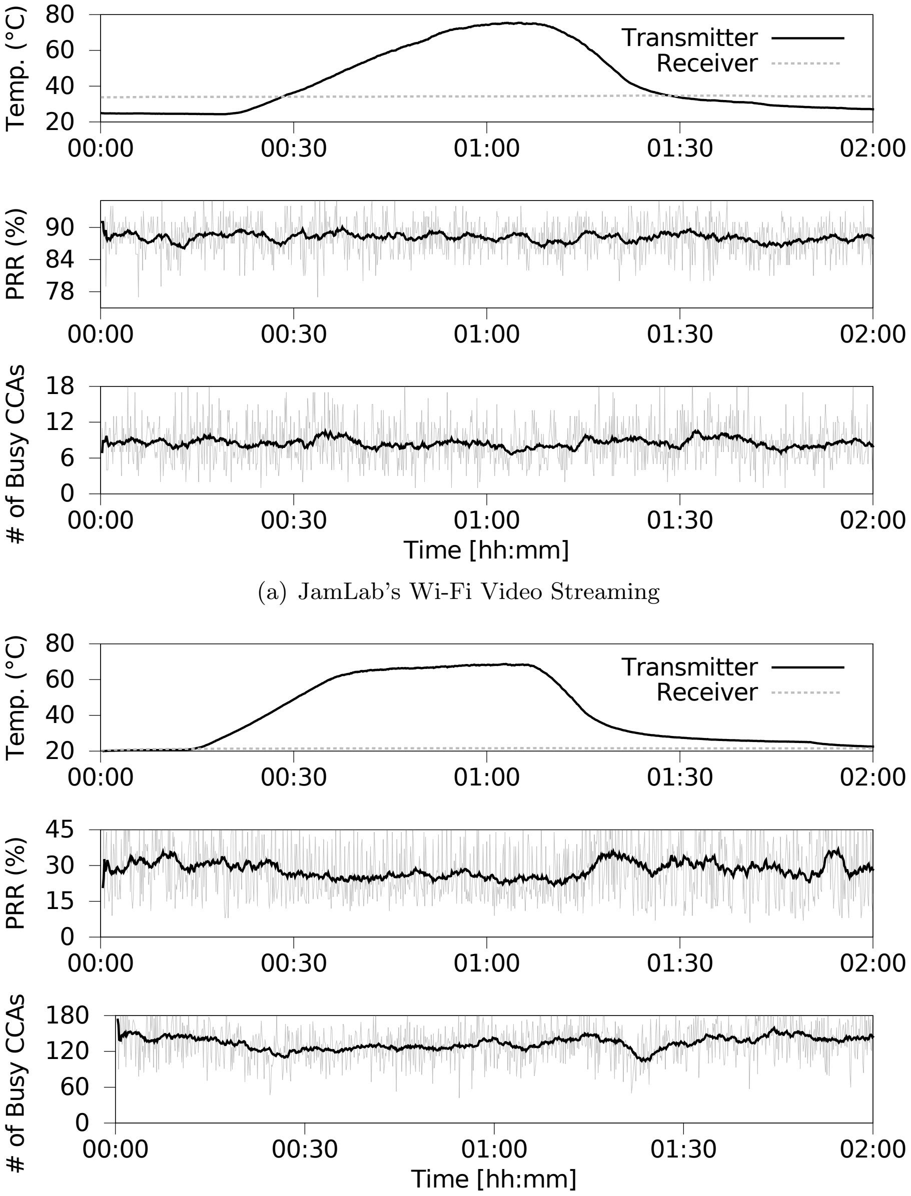
We have evaluated the performance of our adaptive technique experimentally. First, we have shown that it alleviates the collision avoidance and the wake-up problem in CSMA protocols. Second, we have shown that when employing a MAC protocol with an adaptive threshold, the performance of the network significantly increases, with up to 42% lower radio duty cycle and 87% higher PRR in the presence of temperature variations commonly found in outdoor deployments.

Improved Collision Avoidance. Figure 3.20 shows the PRR experienced by the links in the same interference scenarios described in Section 3.3.2 (the experiments were executed back-to-back). If we compare the results with Figure 3.17, we can notice that the PRR does not depend on the on-board temperature of the nodes, but remains instead fairly constant throughout the experiment. This shows that the adaptive protocol can avoid the intersection between the RSSI curve and T_{CCA}, actually mitigating the collision avoidance problem.

Improved Wake-Up Efficiency. We compare the performance of (i) an unmodified ContikiMAC using a fixed CCA threshold, (ii) an adaptive threshold based on local temperature information, and (iii) an adaptive threshold based on the information inferred from the routing layer, in the presence of a network with high temperature dynamics. Figure 3.21 shows the PRR on a representative link in our testbed (a similar trend was observed across all links): the adaptation of the CCA threshold can significantly alleviate the wake-up problem. When using a fixed CCA threshold, the link starts to experience packet loss at 31°C. Instead, the link sustains 100% delivery rate up to 40°C when using local temperature information and up to 64°C when using the information inferred from the routing layer. This essentially implies that the use of a dynamic T_{CCA} extends the usability of a link to a higher temperature.

It is important to highlight that the adaptation of T_{CCA} does not mitigate completely the impact of temperature. The reason lies in the selection of $T^{'}_{CCA}$: by selecting

[8] A Destination-Oriented DAG (DODAG) is a DAG rooted at a single destination, i.e., at a single DAG root (the DODAG root) with no outgoing edges [225].



(a) JamLab's Wi-Fi Video Streaming

(b) Real Wi-Fi File Transfer

Figure 3.20: When adapting the CCA threshold based on local temperature measurements, temperature does not affect the efficiency of collision avoidance in CSMA protocols. In contrast with the results shown in Figure 3.17, the PRR remains fairly constant for all interference scenarios despite temperature variations [41].

$K = 6$ dBm, the high temperature variation attenuates the signal strength by several dB, reaching the physical limit of the radio (i.e., at temperatures higher than 64°C we receive a signal strength that is too weak to be successfully demodulated by the CC2420 radio). Hence, the higher is K, the higher can be the performance gain compared to a protocol using a fixed CCA threshold.

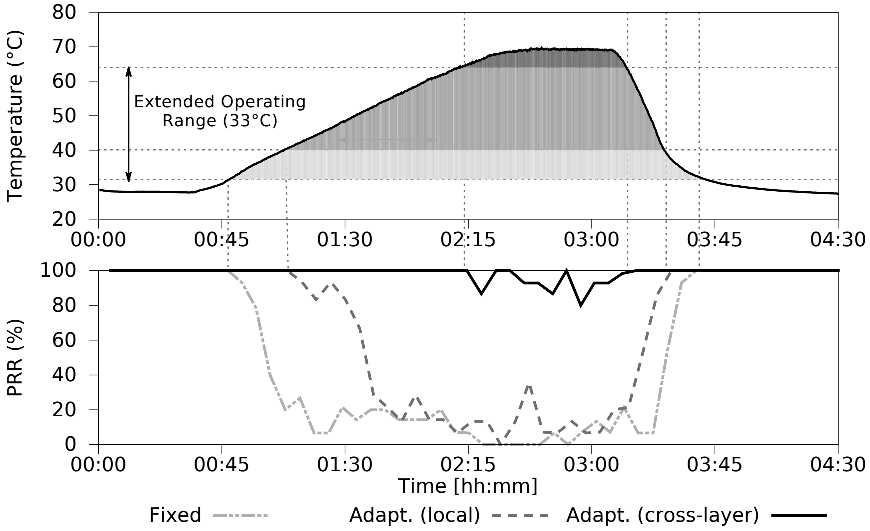


Figure 3.21: By adapting the CCA threshold, we can extend the usability of a link at much higher temperatures, and alleviate the wake-up problem significantly [41].

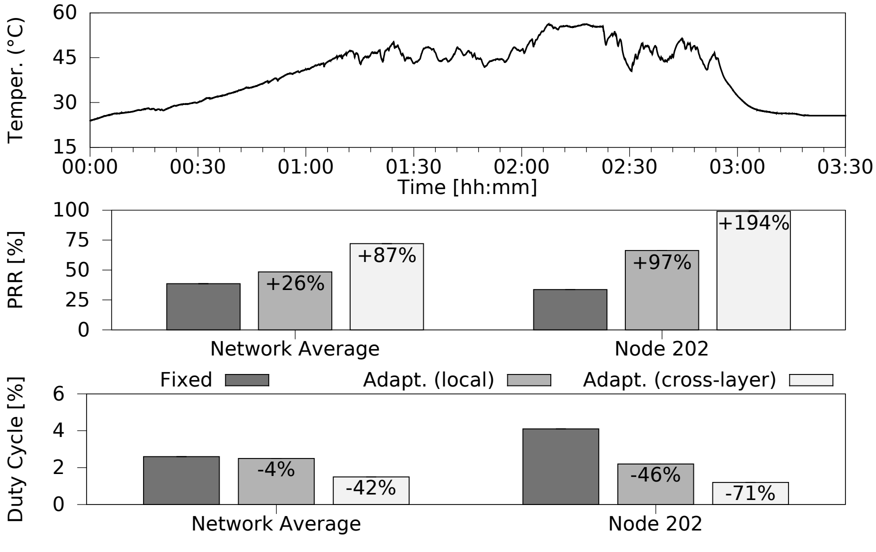


Figure 3.22: Regeneration of a real-world outdoor temperature trace and impact of temperature changes on PRR and duty cycle on a network level and on a single node [41].

Performance on a Network Level. We finally use TempLab to study the performance improvements introduced by our adaptive approach on a network level. Our results indicate that adapting the CCA threshold dynamically in the presence of temperature variations significantly improves performance, especially in sparse networks. If the network is dense, the routing layer can mitigate the impact of temperature and sustain a high PRR even with a MAC protocol employing a fixed CCA threshold (thanks to the ability of the network layer to select alternative links).

The less dense the network is, the higher becomes the impact of temperature on a protocol using a fixed threshold, with the average PRR in the network dropping below 50%. Instead, when using adaptive thresholds, the network sustains higher reception rates in sparse networks (from 44 to 63%, and from 57 to 81% in the two sparsest configurations), with the highest PRR recorded when using the information inferred from the routing layer. We further analyse the energy-efficiency of the different approaches by comparing the average radio duty cycle in the network. In the sparsest network configuration, the duty cycle drops from 4.2% to 3.2% in the case of local temperature information and to 2.3% when using the temperature inferred from the routing layer. The latter corresponds to a 55% higher energy-efficiency than when using a fixed threshold. With denser networks the duty cycle decreases, as the network layer can select alternative links and seamlessly mitigate the impact of temperature.

We finally use TempLab to time-lapse a 24-hours trace recorded in an outdoor deployment [220], and study the impact on a network with a density of one node every 8 m^2 ($T^{'}_{CCA} = n^{'}_{f} + 6$ dBm). The results show that the adaptive approaches that we proposed significantly improve performance, both on a link basis and on a network level. Figure 3.22 shows that the network sustains up to 42% lower radio duty cycle and 87% higher PRR in the presence of temperature variations commonly found in outdoor deployments, and that a single link may experience even up to 71% lower duty cycle and 194% higher packet reception rate.

Chapter 4

Radio Interference

This chapter describes the contributions of this thesis with respect to the impact of radio interference on WSN performance. Section 4.1 describes how to measure and characterize interference using off-the-shelf motes, and provides lightweight models that can be used to make WSN protocols interference-aware. Section 4.2 describes JamLab, a low-cost extension for WSN testbeds that allows to reproduce interference patterns similar to the ones produced by common wireless devices. Section 4.3 analyses the impact of radio interference on the performance of MAC protocols, and identifies a number of mechanisms that can improve their reliability. These mechanisms are embedded within an existing X-MAC implementation, improving its packet delivery rate and energy-efficiency in the presence of interference. Section 4.4 focuses on the agreement problem in congested environments, and illustrates the design and implementation of JAG, a protocol that uses a jamming sequence of configurable length as a last iteration of a handshake to make sure that two neighbouring nodes agree on a given piece of information.

4.1 Interference Characterization and Modeling

In the presence of external interference, the properties of a wireless channel can change unpredictably over time [245, 246]. Interference can be sporadic, causing only a temporary impact on communications, or persistent, causing a channel to experience heavy interference and become unavailable for long periods of time. Wireless sensor nodes may therefore need to adapt dynamically to changing interference patterns and adjust their behaviour at runtime in order to maximize the reliability of their communications.

To achieve this goal, wireless sensor nodes firstly need to acquire a detailed understanding about the surrounding interference by means of accurate measurements. The latter must be carried out in a simple and energy-efficient fashion, in order to meet the constrained capabilities of wireless sensor nodes. Using runtime interference measurements, sensor nodes can then parametrize lightweight interference models and carry out a dynamic protocol selection or a dynamic adjustment of protocol parameters as soon as certain properties in the environment have changed.

In this section, we describe how to accurately measure interference using off-the-shelf wireless sensor nodes (Section 4.1.1), and show how these measurements can be used to study the characteristics of common interference sources in the 2.4 GHz band (Section 4.1.2). We then describe simple interference models that can be implemented on resource-constrained wireless sensor nodes (Section 4.1.3), and explain how they can be parametrized at runtime to quantify the congestion on a channel or to achieve an effective interference mitigation.

4.1.1 Measuring interference accurately using motes

Link quality indicators such as RSSI and LQI provide an indication of the signal strength and quality, but only upon the reception of a packet. Together with the packet reception rate, these metrics can be used to derive the presence of interference and react accordingly (e.g., by switching channel when high packet losses occur [185]), but do not unequivocally identify nor quantify the presence of interference in the environment. For example, the LQI is a measure of the chip error rate and has a correlation with the amount of interference in a given environment, but low LQI values may also result from unreliable links in a complete absence of external interference.

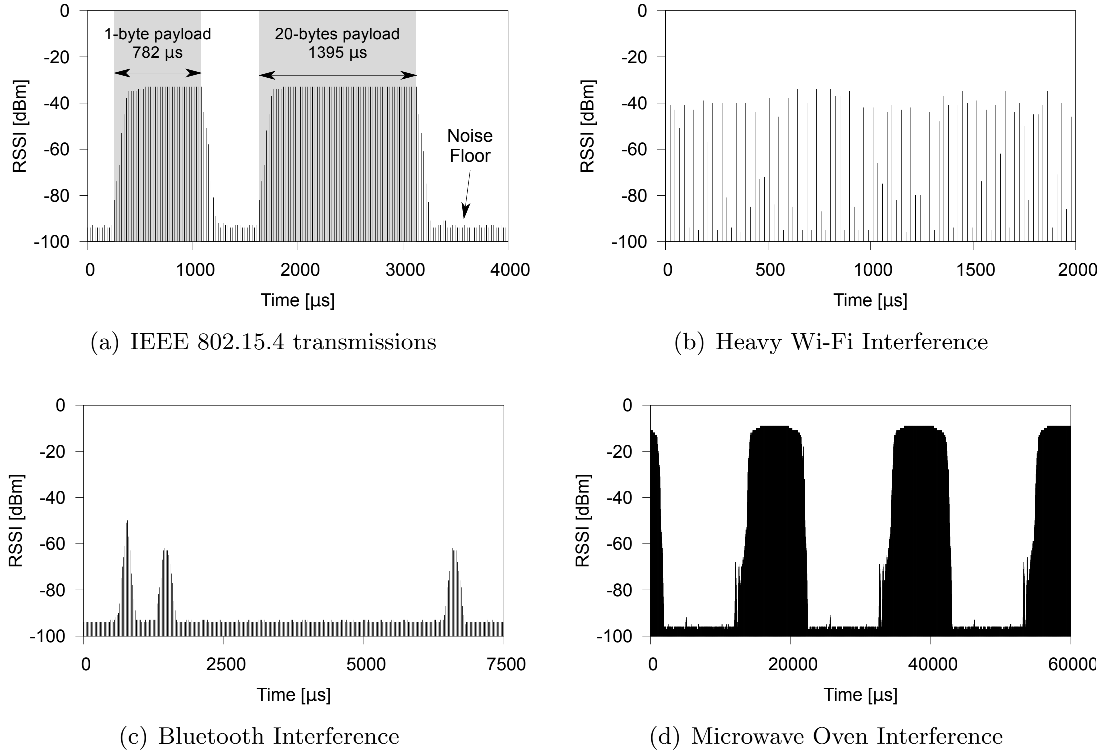
To obtain a snapshot of the ongoing activity in the channel, one can resort to the continuous sampling of the energy level in the wireless channel, also known as RF noise measurement [38]. RF noise measurements are typically retrieved using the energy detection feature available in IEEE 802.15.4-compliant radios, which provides an RSSI value in dBm that corresponds to the strength of the interfering signal (if any) at the antenna pins. However, carrying out accurate RF noise measurements at a sufficiently high sampling rate to detect short transmission periods such as the ones generated by Wi-Fi devices is a challenge due to limited resources available in common wireless sensor nodes.

Measuring at high sampling rates. We improve existing Contiki tools [2] and develop an RSSI scanner carrying out RF noise measurements on a single channel with a maximum sampling rate of approximately 60 kHz in nodes employing the MSP430 microcontroller and the CC2420 transceiver. To achieve this high frequency, we optimized SPI operations, and compressed the buffered RSSI readings using run-length encoding[1].

A sampling rate of 60 kHz is sufficient to detect IEEE 802.11b frames, but is not enough to capture all 802.11g/n frames (the minimum size of a Wi-Fi packet is 38 bytes, and the maximum speed of Wi-Fi transmissions is 11, 54, and 150 Mbit/s for 802.11b/g/n standards, respectively). However, since most Wi-Fi frames are data frames and typically contain higher layer headers, and since the IEEE 802.11n standard uses large PDUs to reduce preamble overhead, we can still capture a significant fraction of Wi-Fi traffic [42].

Avoiding saturation in RSSI readings. Our experiments have highlighted that the accuracy of the obtained RSSI readings degrades when pushing the performance of common sensor nodes to the edge. In particular, several RSSI readings captured at high sampling rate are significantly below the supported range and the sensitivity threshold of the radio (e.g., -110 or -115 dBm). This occurs systematically in three specific scenarios: (i) when a narrowband unmodulated carrier is transmitted, (ii) when microwave ovens are switched on, and (iii) in the presence of Bluetooth transmissions. We experimentally identified that the problem is due to the saturation of the intermediate frequency amplifier chain, and observed that maximum gain is used in the Variable Gain Amplifier (VGA) when the incorrect RSSI readings occur.

[1] Full details about the implementation of the high-frequency RSSI sampling can be found in Paper C included in this thesis [42].



(a) IEEE 802.15.4 transmissions

(b) Heavy Wi-Fi Interference

(c) Bluetooth Interference

(d) Microwave Oven Interference

Figure 4.1: High-frequency RF noise measurements obtained using Maxfor MTM-CM5000MSP sensor nodes in different environments [46].

To linearise the radio response for an arbitrary noise signal and hence avoid wrong RSSI readings, we activate the peak detectors in-between the amplifier stages so that their output is used by the AGC algorithm to compute the required gain.

4.1.2 Characteristics of common interference sources

We now use high-frequency RF noise measurements to study common sources of interference. When neither interference nor IEEE 802.15.4 communications are present, the RF noise measurements typically return RSSI values in the proximity of the radio sensitivity threshold (e.g., in the range $[-100, -94]$ dBm for the CC2420 radio). In the presence of IEEE 802.15.4 communications, the RSSI measurements return a stable value corresponding to the strength and the length of the transmitted packet (Figure 4.1(a)). As packets have a constrained maximum payload size of 127 bytes according to the 802.15.4 PHY standard, a packet transmission at 250 Kbit/sec would not last more than 4.3 ms.

When other devices operating in the same frequency band of wireless sensor networks are active, bursts of interference signals (*busy periods*) alternate with instants in which the channel is clear (*idle periods*). The strength of the interference signals and the duration of idle and busy periods depend on the interfering source and on the specific context. Some devices, such as microwave ovens, generate periodic interference patterns with relatively long idle periods, while others, such as Wi-Fi stations, generate interference patterns with short idle periods of a highly variable length.

Figure 4.1(b) shows an example of interference measurements obtained in the presence of heavy Wi-Fi interference (caused by a file transfer between an access point and a laptop). As it is possible to identify RSSI values matching the radio sensitivity threshold (i.e., noise floor) between consecutive Wi-Fi transmissions, the resolution of the scanner is sufficiently high to identify the Inter-Frame Spaces (IFS) between IEEE 802.11 packets.

Figure 4.1(c) shows an interference measurement taken in the presence of two Bluetooth devices exchanging a file. The characteristics of Bluetooth interference are rather different from the ones of Wi-Fi, as the protocol uses an adaptive frequency hopping (AFH) mechanism to combat interference, and as it hops among 1-MHz channels around 1600 times/sec. Since Bluetooth channels are more narrow than the ones defined by the 802.15.4 standard, it may happen that communication in multiple adjacent Bluetooth channels affects a single 802.15.4 channel.

The interference generated by microwave ovens is rigorously periodical, as shown in Figure 4.1(d). Microwave ovens emit high-power RF noise in the 2.4 GHz ISM band in a periodic fashion, and the period in which the oven is active depends on the power grid frequency. Figure 4.1(d) shows a period of roughly 20 ms, which matches our expectations (literature reports a power cycle of roughly 20 ms (at 50 Hz) or 16 ms (at 60 Hz) with an active period of at most 50% of the power cycle [42, 114]).

4.1.3 Lightweight interference models

The primary implication of radio interference is packet loss. The latter occurs in the presence of a sufficiently strong interference signal, such that the receiver node is no longer able to discriminate the good signal from the interfering one. The receiver node can indeed reject any interference that is C_{Rej} weaker than the signal of interest, with C_{Rej} being the so called co-channel rejection capability of the transceiver (with unit dB). Any interfering signal stronger than that may result, depending on its duration and strength, in either a corrupted or a lost packet. The first case occurs when radio interference corrupts only some of the bits in a frame, leading to Cyclic Redundancy Check (CRC) errors and a consequently dropped packet. The second case (which corresponds to the vast majority of the cases) occurs when the radio does not even detect the presence of a frame.

Radio interference can also cause large energy expenditures when using low-power duty-cycled MAC protocols, as it would trigger unnecessary wake-ups in the presence of interfering signals stronger than the CCA threshold in use [187]. All these scenarios can be modelled without knowing the actual strength of the interfering signal. It is in principle sufficient to know only whether the interfering signal is above or below a given value (e.g., the CCA threshold in use, or the signal strength with which packets are received).

Channel occupancy model. We can therefore model interference using a two-state channel occupancy model [194], in which, at a given time instant, a channel is defined as busy if any interfering signal is above a threshold R_{thr} and defined as idle otherwise [243]. Denoting x_i as the RSSI measured by a node at a given time instant, the occupancy of the channel can be expressed as:

$$X_i = \begin{cases} Busy \ \ (1) & if \ \ x_i > R_{thr} \\ Idle \ \ (0) & if \ \ x_i \leq R_{thr} \end{cases} \tag{4.1}$$

with X_i being a binary number specifying a busy channel (1) or an idle channel (0).

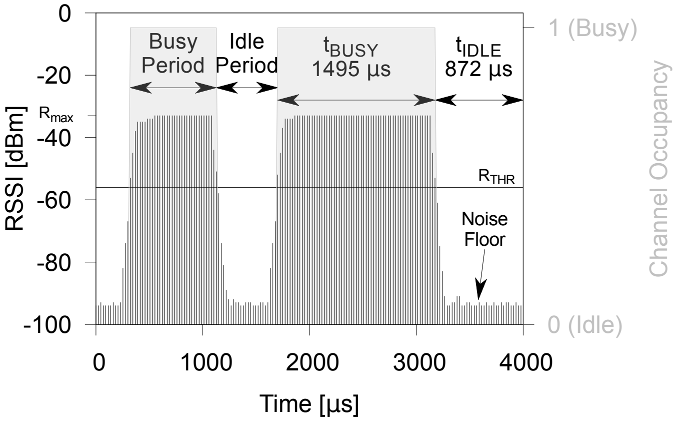


Figure 4.2: Interfering signal of strength R_{max} recorded by a Maxfor MTM-CM5000MSP node performing a high-frequency RF noise measurement. The shaded area shows the busy period of interference according to the two-state channel occupancy model [243].

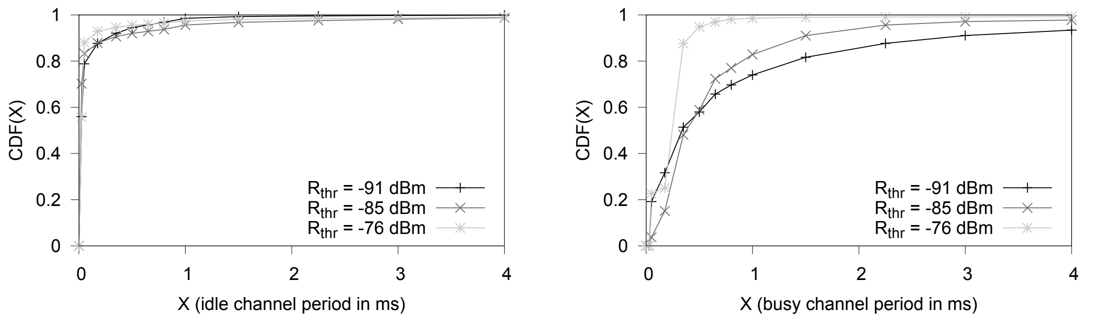
The advantage of this model is its simplicity: it can be easily used on constrained sensor nodes that are able to carry out RF noise measurements. Figure 4.2 shows an example in which a Maxfor MTM-CM5000MSP node is continuously sampling RSSI values at high frequency using the technique illustrated in Section 4.1.1. Denoting $\{x_1, x_2, \ldots, x_n\}$ as the sequence of consecutive RSSI values sampled at a rate of R Hz, and $\{X_1, X_2, \ldots, X_n\}$ as the binary sequence of channel occupancy states computed according to Equation 4.1, one can derive an alternating sequence of idle and busy periods. Their duration can be computed by knowing how many consecutive RSSI values are above or below R_{thr}.

Distribution function of idle and busy periods. In principle, the longer the idle period, the higher the likelihood that a packet will be successfully received. For several protocol parameters, such as the CCA back-off time between consecutive busy channels [43], or the payload length [47], it is often important to know the actual distribution of idle and busy periods. Figure 4.3 shows an example of the Cumulative Distribution Function[2] (CDF) of idle and busy periods measured by a Maxfor MTM-CM5000MSP node in the presence of a laptop continuously downloading a file from a nearby access point [46]. In such a scenario, the probability of having an idle period longer than 2 ms is smaller than 5%. A MAC protocol should hence use as short payloads as possible, as well as avoid long CCA back-off times. On the contrary, an environment in which interference occurs in long bursts with large idle periods would call for large payloads in order to minimize the number of packet transmissions (i.e., the amount of time in which the radio is on).

Denoting $p_i(i)$ as the probability density function (*pdf*) of the idle periods formed by the interference pattern, a protocol could for example select the optimal payload length [47] by computing the probability of encountering an idle period of length i:

$$s(i) = \frac{ip_i(i)}{\sum_{i=1}^{\infty} ip_i(i)} \tag{4.2}$$

[2] The CDF of a random variable X is the function given by $F_X(x) = P(X \leq x)$, where the right-hand side represents the probability that the random variable X takes on a value less than or equal to x.



(a) Idle periods

(b) Busy periods

Figure 4.3: CDF of idle and busy periods measured using different R_{thr} thresholds in the presence of a laptop continuously downloading a file from a nearby Wi-Fi access point [46].

Similarly, denoting $p_b(i)$ as the probability density function of the busy periods, a protocol could for example select the optimal back-off time [43, 243] for clear channel assessment by knowing the probability of selecting a busy period of length i:

$$t(i) = \frac{ip_b(i)}{\sum_{i=1}^{\infty} ip_b(i)} \tag{4.3}$$

To measure the distribution of idle and busy periods, we use a modification of the measurement tool described in Section 4.1.1. We continuously scan the radio channel and detect whenever the activity in the channel is above a configurable threshold R_{thr}. We then map the number of consecutive RSSI samples above or below R_{thr} to the time in which the medium remained busy or idle, respectively [46]. To address the resource limitation of sensor nodes, we discretize the retrieved CDF and use lookup tables to make sure that the execution time between RSSI readings remains approximatively constant.

Modelling the congestion on a channel. The two-state channel occupancy model can also be used to quantify the congestion of a channel. Several works have used the average energy in the channel as an indicator of its usage [145]. However, such metric is unable to distinguish between a usable channel and a noisy channel. An interference source generating an intermittent bursty traffic would still allow successful transmissions between bursts, whereas an interfering device streaming continuous media traffic would significantly lower the probability of successful transmission.

Let m_j denote the number of idle periods consisting of j consecutive idle RSSI samples, n the total number of RSSI samples, and $m_1 + m_2 + \ldots + m_n = m$ the total number of idle periods. Assuming that the RSSI is sampled at a frequency $1/P$, j consecutive clear samples hence imply that the channel was free for at least $(j-1) \cdot P$ time units. One can define the average channel vacancy CV as:

$$CV(\tau) = \frac{1}{n-1} \sum_{j|(j-1)P>\tau} jm_j \tag{4.4}$$

where $\tau > 2P$ is the time window of interest, which could be mapped to the time in which a packet stays over-the-air. Depending on the user needs, one can bias Equation 4.4 to rank channels with larger vacancies higher as in [149], and define customized channel quality metrics that are agnostic to the interference sources in the surroundings.

4.2 JamLab: Realistic Experimentation with Interference on WSN Testbeds

In order to analyse and compare the performance of communication protocols in the presence of common interference patterns, we have designed JamLab, a low-cost extension for WSN testbeds enabling easy experimentation with radio interference [42][3]. JamLab supports the recording and playback of interference traces in a testbed without the need of additional hardware, as well as the customizable generation of typical interference patterns caused by devices operating in the frequency of interest.

Reproducing radio interference on large-scale testbeds without resorting to expensive hardware can be quite challenging, as it is not sufficient to randomly transmit packets within the testbed to interfere the ongoing communications. JamLab's design has indeed been driven by a number of requirements that we describe in Section 4.2.1.

4.2.1 Requirements

When testing the reliability of a protocol or an application against external interference in a systematic fashion, one needs to set up realistic and credible experiments [84]. The interference patterns used in the experiments must be a representative set of how interference appears in reality. Having a device that is permanently interfering for long periods of time would not represent a realistic scenario, as it hardly occurs in practice (interference is instead typically bursty). The testbed infrastructure should hence faithfully reproduce interference patterns that can be found in the real-world (ideally, it should recreate an environment similar to the one at the target deployment location).

A testbed with the ability of reproducing interference patterns should hence be able to satisfy the following requirements:

- *Impact on network performance.* The interference reproduced within the testbed should be able to affect network performance in the same way as interference in real-world deployments does. Hence, when appropriate, the interfering signals should be sufficiently strong to block the communication between two sensor nodes and generate packet loss. Similarly, the impact on energy efficiency caused by retransmissions and false wake-ups in duty-cycled protocols should also be faithfully reproduced.

- *Temporal accuracy.* The testbed should have the ability of reproducing interference patterns over time in a fine-grained fashion. For example, when reproducing the interference caused by a Wi-Fi device, one should have the ability of varying its patterns as a function of the user activity, the transport protocol, the packet size, or other low-level parameters. Similarly, when reproducing the interference caused by a microwave oven, one should be able to precisely tweak the duration of a power cycle depending on the power grid frequency in use or on the model of the device.

- *Spatial accuracy.* Radio interference typically affects only portions of the network, especially if the latter is spread over large distances. The testbed should hence have the ability to recreate different interference patterns across the network.

[3] Full details about JamLab's design and implementation can be found in Paper C included in this thesis [42].

- *Frequency diversity.* As there is an increasing trend to use multiple IEEE 802.15.4 channels in order to increase the robustness to interference and the overall bandwidth (see Section 2.3.2), the testbed should provide the ability to reproduce interference in multiple channels. This should be done in consideration of the actual characteristics of common wide-band interferers such as Wi-Fi access points and microwave ovens.

- *Repeatability.* A fundamental requirement when comparing or debugging the performance of different protocols is the ability of repeating an experiment under the same conditions, i.e., the testbed should reproduce the same interference patterns and affect the nodes in a similar way across multiple experiments.

- *Scalability and controllability.* The infrastructure for interference generation should ideally affect the entire network without requiring an excessive amount of time and resources. All interfering devices should be remotely controllable: activating several devices manually (e.g., microwave ovens) in a large-scale network would be infeasible.

- *Low cost.* All the above requirements have to be satisfied while minimizing the cost of the solution.

4.2.2 Architecture

The key idea behind JamLab is to use off-the-shelf motes to record and playback interference patterns instead of bringing Wi-Fi access points, microwave ovens, or other equipment to the testbed. The latter approach is not only costly and hard to reproduce by other researchers, but it is even difficult to exactly reproduce a given interference pattern with the same appliance. For example, the sequence and timing of the Wi-Fi frames generated by a file download may differ between repeated trials due to TCP adaptation mechanisms and due to the amount of traffic in the network infrastructure. Furthermore, every device used to generate interference in the testbed needs to be programmed remotely. Programming several heterogeneous devices such as Wi-Fi access points or microwave ovens would create a significant overhead, whereas using off-the-shelf sensor nodes the installation overhead is minimal (a simple software upload), making life easier for research groups not equipped with sophisticated test instruments to generate realistic controlled interference.

With JamLab, either a fraction of the existing nodes in a testbed are used to record and playback interference patterns, or a few additional motes are placed in the testbed area. We call those motes used for interference generation *HandyMotes*. The HandyMotes support two modes of operation:

- *Emulation*, where the lightweight models derived in Section 4.1.3 are used to generate interference patterns that resemble those generated by a specific appliance (e.g., Wi-Fi device and microwave ovens);

- *Regeneration*, where each HandyMote autonomously samples the actual interference, compresses and stores it locally, and regenerates the recorded patterns at a later stage.

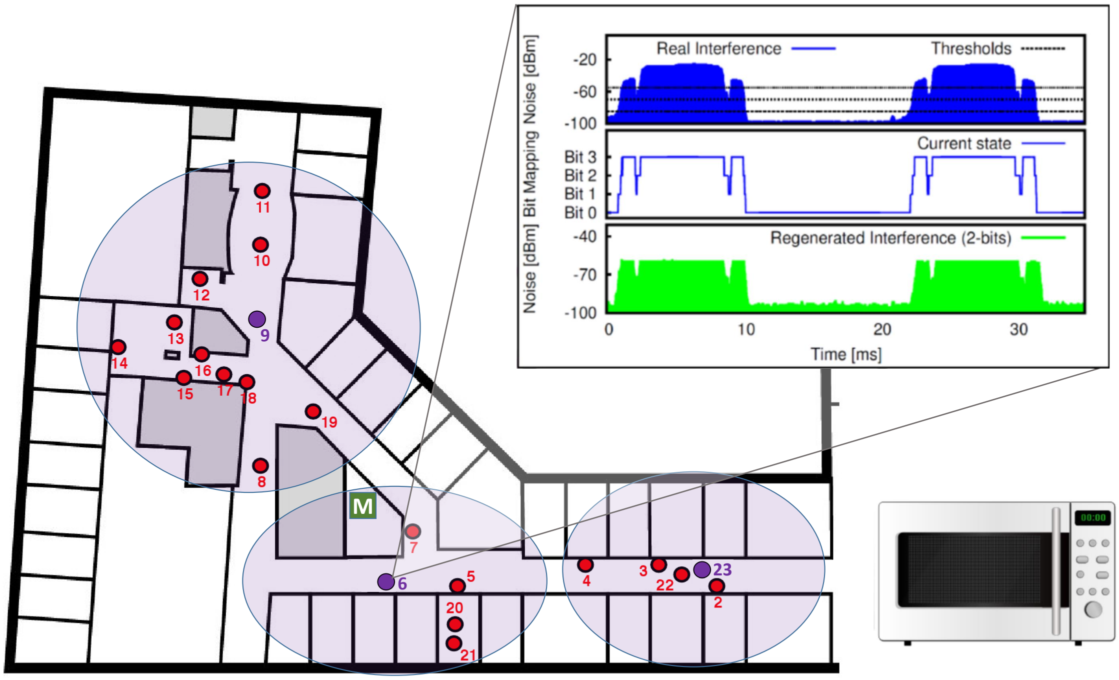


Figure 4.4: Overview of JamLab's architecture.

The latter mode is especially useful to record realistic interference patterns in a crowded shopping center or on a lively street by placing a few HandyMotes to record interference, and bringing them to the testbed to playback the recorded traces there.

One fundamental challenge results from the fact that the maximum RF output power of motes (0 dBm) is typically much smaller than the RF output of other typical interference sources (approx. 25 and 60 dBm for Wi-Fi and microwave ovens, respectively). Therefore, a Wi-Fi transmitter or a microwave oven may disturb WSN communications over much larger distances than a HandyMote can. We address this issue by subdividing the testbed area into cells as depicted in Figure 4.4, such that a HandyMote placed at the center of the cell can interfere with all testbed motes contained in the cell, but the interference with motes outsides of the cell is minimized. This requires a careful placement or selection of the HandyMotes and control of their RF output power, as explained in Section 4.2.4. Note that there is a trade-off between the realism of the generated interference patterns and the number of HandyMotes: the more cells, the more accurate is the spatial distribution of interference, but the more HandyMotes are required.

A second fundamental challenge is the recording and playback of interference using off-the-shelf motes. On the one hand, one needs high sampling rates with low jitter to capture short interference patterns such as those generated by Wi-Fi beacons, as well as accurate measurement of the interfering signal strength (the solution to these problems was illustrated in Section 4.1.1). On the other hand, to playback the recorded interference traces, normal packet transmissions are not appropriate, as this would offer only limited control over the exact timing of the transmitted signals. We therefore propose to use special test modes of IEEE 802.15.4 radios to generate modulated or unmodulated carrier signals as detailed in Section 4.2.3. Furthermore, off-the-shelf radios only offer a limited number of discrete output power levels that can be exploited to control the generated interfering signal strength.

The minimum requirement is that each HandyMote can produce *binary interference*, i.e., that it can block the communication of the motes in its cell by emitting a strong-enough interference signal, or that it does not interfere at all when inactive. The remaining output power levels supported by the radio hardware are then used to fine-tune the strength of the interfering signal accordingly.

A third fundamental challenge is that many interference sources emit wideband signals, i.e., they interfere with many IEEE 802.15.4 communication channels at the same time. In contrast, a mote can only transmit on a single channel at a time. To deal with this issue, one can place multiple HandyMotes in each cell, each one interfering on one IEEE 802.15.4 channel. To synchronize the generation of interference patterns within the HandyMotes in one cell and across cells, one can use the wired back-channel of the testbed infrastructure to send synchronization signals to the HandyMotes.

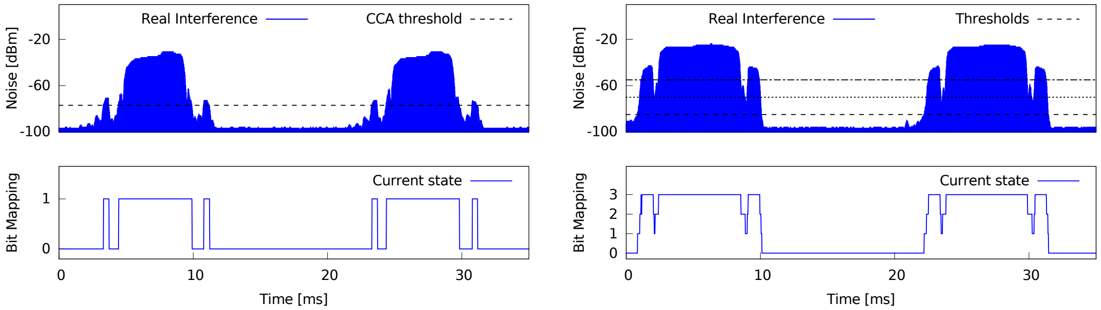
4.2.3 Implementation

JamLab has been designed specifically for the CC2420 radio [206], and tested on several sensor motes such as Maxfor MTM-CM5000MSP, Crossbow TelosB, and Sentilla JCreate. However, the same framework can be easily applied to other WSN platforms (the port would be almost immediate for sensor nodes equipped with EM2420 and CC2520 transceivers).

We have developed the HandyMotes using Contiki [72], and used the techniques to accurately measure interference introduced in Section 4.1.1 to record and replay those patterns. We now describe how to compress and store the measured interference traces on motes and how to playback those recordings.

Recording interference traces. When used in *regeneration* mode, a HandyMote needs to record interference traces that will be played back in the testbed at a later stage. Those traces can be either stored on the mote in RAM or flash memory, or – if the HandyMote is connected to a testbed during recording – can be streamed over a wired back-channel to a base station. In any case, the data rate of 480 kbps generated by sampling RSSI with a resolution of 8 bits to hold values between 0 and -100 dBm at 60 kHz is too high to store it directly in memory or to stream it over the back-channel. The very efficient Coffee flash file system supports a peak write bandwidth of only 376 kbps [210], the MSP430 UART supports a maximum data rate of 460 kbps for writing to the USB back-channel, and the limited 4 kB RAM of the MSP430 could just record a trace of less than 70 ms duration.

While we need a high compression ratio, the compression method has to be efficient enough to allow sampling of RSSI at 60 kHz. Therefore, we use a simple run-length encoding strategy and a quantization of the samples to a few bits per sample. We store a stream of pairs (v, o), where v is a sample and o is the number of consecutive occurrences of this sample. This method is very effective, as RSSI values typically change slowly over time. The quantization is justified by the fact that the CC2420 only supports 11 distinct output power levels in the range [-55,0] dBm. To obtain the highest possible output resolution, four bits per sample with an appropriate non-linear quantization are



(a) 1-bit quantization

(b) 2-bit quantization

Figure 4.5: Encoding techniques to save memory resources when recording a trace [42].

hence sufficient. For example, for two-bit resolution one can use thresholds -55, -70, and -80 dBm, for quantizing the RSSI range into four regions.

The two-bit quantization of a 35 ms interference recording reduces the amount of data from 2076 bytes to 84 bytes – a compression ratio of 25:1. Figure 4.5(b) shows how RSSI readings recorded in the presence of an active microwave oven (top) are mapped into 2 bits (bottom). To support binary interference regeneration, even a single bit can be sufficient: this would correspond to the outcome of a continuous CCA operation. Figure 4.5(a) shows the outcome of a one-bit quantization of 35 ms of interference: the amount of data is reduced from 2076 bytes to 20 bytes – a compression ratio of almost 1:100. The compression and quantization of the traces hence significantly reduces the data rate to a level that can be handled by flash and USB, and even allows us to store recordings up to tens of seconds or minutes in RAM.

Generating interference using off-the-shelf motes. To playback the recorded interference traces, normal packet transmissions are not appropriate, as this approach would offer only limited control over the exact timing of the transmitted signals. Therefore, we use special test modes of IEEE 802.15.4 radios to generate modulated or unmodulated carrier signals [35, 39] that are stable over time. This approach is superior to common jamming techniques based on packet transmissions, as the emitted carrier signal is independent from packet sizes and inter-packet times.

However, to generate an interference pattern, an interfering node needs to enable and disable the transmitter for a duration that matches the desired interference pattern, as well as set its output power accordingly. Enabling the transmitter using the STXON command is not a suitable option, as the radio oscillator first has to stabilize before a transmission is possible, resulting in a latency of 192μs or a maximum playback frequency of only 5 kHz. We therefore leave the HandyMote on all the time and use the PA_POWER level 0 (≈ -55 dBm) instead of disabling the transmitter (the RF output power at this level is so low that even a receiver at a distance of only few centimetres can hardly detect the signal). The advantage of this approach is that the latency for changing the output power is now dominated by the SPI access time. The SPI optimization in Contiki that was implemented to measure RSSI values at high speed results in a latency of only few microseconds – allowing us to to playback at the same frequency of 60 kHz that was also used during recording.

Model-based emulation of interference patterns. An alternative to the *regeneration* of interference patterns is the *emulation* of the interference produced by specific devices operating in the frequency of interest. We use the channel occupancy model described in Section 4.1.3, to derive the cumulative density function of several interference sources by means of empirical channel measurements. We then build a library of interference patterns produced by common wireless devices, and record interference in different scenarios (e.g., we vary the number of users or the type of traffic when recording the interference generated by Wi-Fi and Bluetooth devices)[4]. The CDFs of idle and busy periods are then used to randomly generate a binary schedule for the HandyMotes. The latter activate or deactivate the transmission of the carrier signal following the probability density function of idle and busy periods [42].

To emulate the interference produced by microwave ovens we follow instead a different approach. As microwave oven interference is the simplest to model (it follows a deterministic on/off sequence), we define the period of the signal τ, the duty cycle λ (fraction of time in which the oven is 'on'), and hardcode these two parameters into the HandyMote of interest.

4.2.4 Testbed configuration

To deal with the limited RF output power of the HandyMotes, we partition the area of a testbed into different cells (Figure 4.4). Each cell contains a HandyMote and a number of regular nodes that can be used for experimentation, and its size should be selected such that all regular nodes can be interfered by the HandyMote despite its limited RF output power. Furthermore, cross talk between neighbouring cells should be minimized (i.e., a HandyMote should only minimally affect regular motes outside of its cell). Note that there is a trade-off between the size of the cells and the accuracy of the spatial distribution of generated interference: the smaller the cells, the higher is the spatial sampling resolution and the smaller are the cross-talk regions. However, smaller cells also imply that more HandyMotes are needed to cover the testbed.

Iterative procedure. We now propose an iterative procedure to find a testbed configuration (i.e., how to select the HandyMotes and how to set their power levels) that maximizes coverage and minimizes cross-talk:

1. In a first step, we obtain from the testbed layout the minimum distance D_{min} between a pair of motes in the testbed. We further derive the minimal signal strength with which a mote can receive a packet P_{min} (this typically corresponds to the CCA threhsold when using duty-cycled MAC protocols, as nodes would not wake up if the signal is weaker than that), and we empirically measure the maximum signal strength P_{max} with which a node receives a message from another mote. The latter can be measured by having all motes in the testbed sequentially broadcast messages while recording the RSSI of received packets.

2. Knowing these parameters, we can compute the maximum cell radius and overlay a hexagonal grid with cells of the computed radius over the testbed layout. We

[4] This library has also been used to augment existing simulation tools with the playback of realistic interference traces [40]. We have extended COOJA [153] and incorporated this library of traces directly into the simulation environment, improving significantly the level of realism.

place HandyMotes roughly at the center of the overlay cells (or select testbed motes close to the center of the cells to become HandyMotes), and allocate motes to the HandyMotes based on the cell overlay.

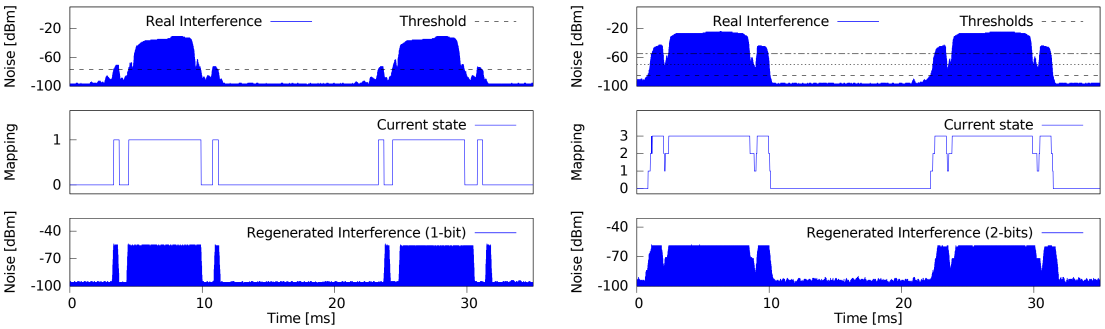
3. Next, we sequentially trigger the selected HandyMotes to generate a continuous interference signal at maximum output power and check if every mote in the cell of a HandyMote is covered. To do this, each regular node can measure the RSSI and check if it is larger than $(P_{max} + C)$ dB. C represents the co-channel rejection threshold of the radio, and is approximately 3 dB for the CC2420 radio. In order to efficiently interfere with the regular nodes in its cell, indeed, a HandyMote needs to produce an interfering signal with a strength that is at least C dB higher than the maximum strength of any other signals that a regular node in its cell may receive.
4. If there are any uncovered motes, we select additional HandyMotes in the vicinity of those motes and return to step 3.
5. In order to reduce cell cross-talk, we reduce the output power levels of the HandyMotes to the minimum value that still guarantees coverage using the same approach as in step 3. If the selected power level is not the maximum power levels, then the power levels higher than the selected one can be used to realize different levels of interference strength. Otherwise, only binary interference can be generated.
6. Finally, one may estimate the quality of the generated configuration by counting the number of motes contained in cross-talk region as follows. The HandyMotes sequentially generate an interference signal at the selected output power. All motes outside of the cell of that HandyMote measure RSSI. If the measured value is larger than $(P_{min} + C)$ dB, then the mote is contained in a cross-talk region. If the number of motes in cross-talk regions is too high, one may start over with a different initial selection of cells.

This iterative procedure is supported by a program running on the testbed motes during the setup phase. After the configuration is completed, the regular nodes can be programmed with the test application. As part of future work, we plan to further automate this procedure.

4.2.5 Evaluation

We now evaluate the accuracy with which a HandyMote can regenerate a previously recorded interference trace in the time domain, and check whether the regenerated interference affects WSN performance in a similar way with respect to the one produced by real interfering devices. We then augment an existing testbed infrastructure with JamLab, and evaluate the accuracy with which the augmented testbed can regenerate a previously recorded interference trace in the spatial domain.

Temporal replay accuracy. We run a HandyMote in regeneration mode in proximity of an active Lunik 200 microwave oven. Figure 4.6(a) (top) shows the interference generated by the microwave as measured by the HandyMote. Next, the trace is quantized to single-bit resolution (middle). Finally, once the microwave oven stopped operating, the HandyMote plays back the recorded binary interference (bottom) using transmission power 0 dBm.



(a) 1-bit quantization

(b) 2-bit quantization

Figure 4.6: JamLab's temporal accuracy in replaying the interference produced by a microwave oven [42].

We quantify the accuracy of a HandyMote regenerating an interference pattern with respect to the originally recorded signal using the *cross-correlation* coefficient (**c**). We represent original and regenerated signals by the series $x(i)$ and $y(i)$, respectively, where $i = 1, \ldots, N$. These series are binary, and take 0 (idle channel) or 1 (busy channel) values. Considering this representation, **c** is given by:

$$\mathbf{c} = \frac{\sum_{i=-\infty}^{\infty} x(i)y(k-i)}{rms(x)rms(y)} \tag{4.5}$$

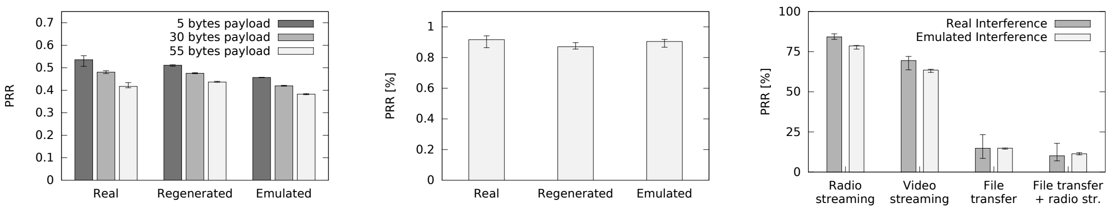
where $rms()$ denotes the root mean square value of a signal. We tested eight pairs of original and regenerated samples and the maximum value of **c** was selected for each pair:

$$\mathbf{c}_{xy} = \max_{k \in [-(N-1),(N-1)]} \{\mathbf{c}\} \tag{4.6}$$

The average correlation $\mathbf{c}_{xy}$ is 0.93 ± 0.065. Hence, our implementation does a commendable job with respect to the cancellation of the jitter between sampled and regenerated interference and hence regenerates interference with a fairly high accuracy. Figure 4.6(b) shows the regeneration process when using a 2-bit resolution, which exhibits a $\mathbf{c}_{xy}$ comparable to the 1-bit resolution.

Impact on network performance. We now evaluate how accurately the interference (re)generated by a HandyMote affects network performance compared to the original interference patterns. First, we measure packet reception between two nodes in several interfered environments, while one HandyMote in the surroundings measures and records interference. We then let the HandyMote reproduce these interference patterns in a non-interfered environment, and compare the loss in performance caused by JamLab's emulated and regenerated interfering signals with the original ones.

We generate interference using (a) the same Lunik 200 microwave oven as in the previous experiment, (ii) two devices transferring a file using Bluetooth, and (iii) a laptop generating different types of Wi-Fi traffic. First, we collect statistics about packet reception on a pair of sensor nodes, in which the sender transmits packets at a rate of 128



(a) Microwave oven

(b) Bluetooth file transfer

(c) Wi-Fi interference

Figure 4.7: JamLab's accuracy in reproducing the same impact on packet reception when regenerating and emulating a specific interference pattern [42].

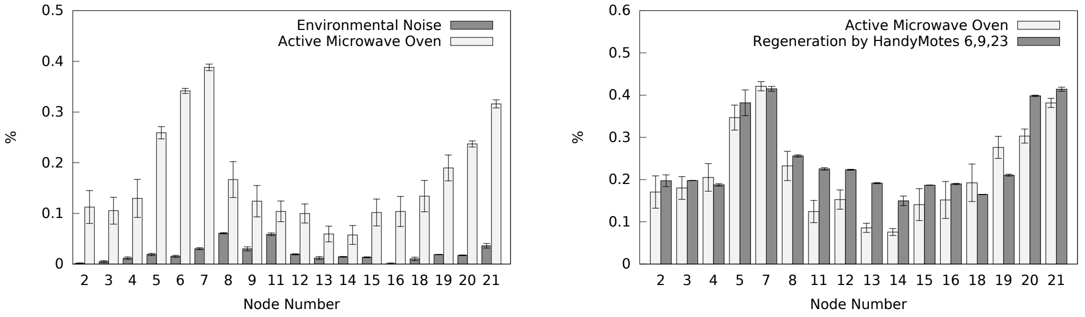
packets/sec, while a HandyMote in close proximity to the two nodes records the interference patterns that are generated. We then place the HandyMote between the two nodes and run it in both emulation and regeneration mode (1-bit quantization). We use the highest transmission power, such that the generated interference signal blocks communication between the sensor nodes.

Figure 4.7 shows the comparison between the packet reception rates across multiple experiments. The HandyMotes do a commendable job in reproducing the loss caused by common wireless devices on a given link. In general, the (re)generated interference produces a slightly lower packet reception rate compared to the original interference (across all the experiments, the average PRR was between 0.25% and 12.83% higher than the original one). The reason behind this is the noisy amplitude of the original interference signal such that occasionally the interference is too weak to block the transmission. In contrast, the interference signal (re)generated by HandyMotes is typically binary and therefore always tends to block communication.

Spatial accuracy. We now study with which accuracy we can regenerate the spatial distribution of interference. For this, we place a Whirlpool M440 microwave oven in the position marked as M in Figure 4.4. We selected a microwave oven as interferer, because of its high output power (which can interfere over long distances) and because of the highly varying strength of the interference produced over time, which makes the regeneration even more challenging.

Our goal is to record the spatial distribution of the interference patterns generated by the microwave through the testbed area. First, we sample and record the interference in channel 23 (which is in our case the most affected) in all nodes. We then let the selected HandyMotes 6, 9, and 23 regenerate the recorded traces throughout the testbed. The regular nodes record the regenerated interference and compare it with the one recorded while the microwave oven was active.

Figures 4.8(a) and 4.8(b) show the amount of interference recorded while the microwave oven was active (a), and while JamLab was regenerating the same patterns (b). We now consider the distribution of the intensity of interference: instead of recording raw traces (as for the temporal evaluation), every mote now computes for which percentage of time the strength of the received signal was higher than P_{max}. The interference regenerated by JamLab is consistent throughout the testbed area. During regeneration, all regular nodes in a cell are interfered with the patterns recorded by the cell's HandyMote. If



(a) Original interference produced by a microwave

(b) Interference regenerated by JamLab

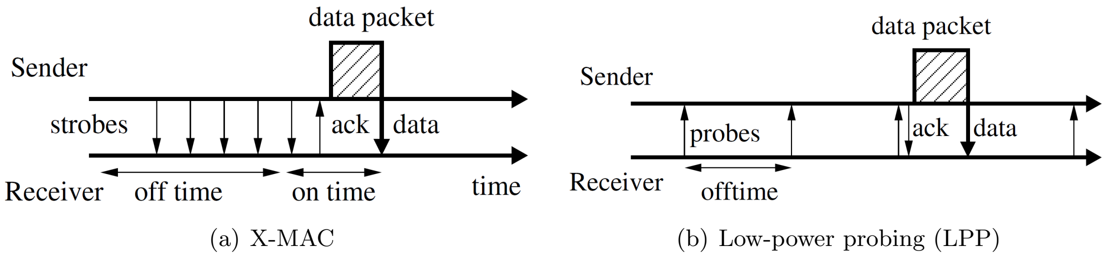
Figure 4.8: Comparison between the interference generated by a microwave oven and the one regenerated by JamLab throughout the testbed area [42].

the latter is further away from the interfering source than a regular node, the replayed interference will be slightly lower than the original interfering source. In the scenario illustrated in Figure 4.4, this is the case for node 19 (further away than HandyMote 9). On the contrary, regular nodes that are further away than a HandyMote with respect to the interfering source will suffer a regenerated interference that is slightly higher than the original one. In our experiments, this was the case for regular nodes 20 and 21 (HandyMote 6 was closer to the microwave) and for regular nodes 13, 12, and 11 (HandyMote 9 was closer to the microwave).

One limitation of JamLab is that the replayed interference "adds" on top of existing background noise in proximity of the testbed. Figure 4.8(a) shows that some nodes receive some additional background interference besides the one produced by the microwave oven. Observing Figures 4.8(a) and 4.8(b), we can see how the interference received by node 8 is higher than the one recorded by HandyMote 9 due to a high environmental noise. In order to reduce the non-determinism caused by differences in ambient interference between recording and regeneration, the experiments should be run when the background noise is low, for example in the evening or during the night.

4.3 Improving Protocol Performance under Interference

We now evaluate the performance of several MAC protocols in congested environments using HandyMotes, and derive techniques to increase their reliability. First, we carry out an experimental comparison of the performance of different protocols, showing that specific features such as hand-shaking schemes preceding the actual data transmission and congestion back-off timers, play a critical role in the presence of interference (Section 4.3.1). Second, building on top of our experimental results, we identify mechanisms that can improve the reliability of existing MAC protocols in the presence of interference, and embed them within an existing X-MAC [49] implementation, showing considerable improvements in the packet delivery rate despite a minimal power consumption (Section 4.3.2).



(a) X-MAC

(b) Low-power probing (LPP)

Figure 4.9: In X-MAC (left), the sender strobes until the receiver is awake and can receive a packet. In LPP (right), the receivers send probes to announce they are awake and ready to receive packets [43]. Preamble and packet durations are not drawn to scale.

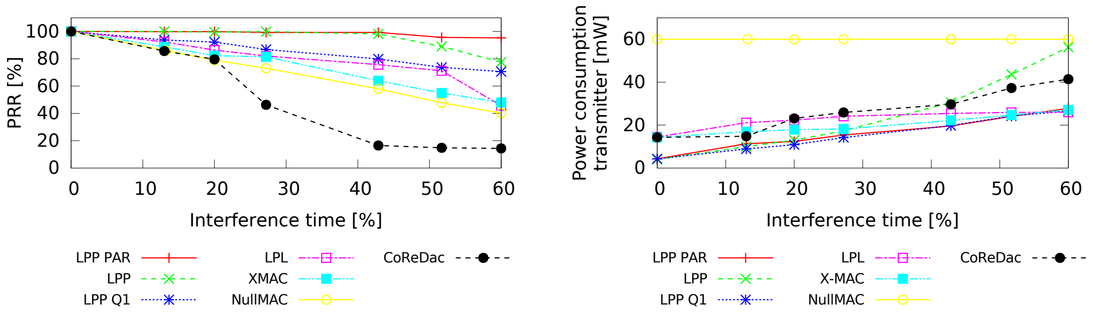
4.3.1 Performance of MAC protocols

We now focus on unicast communications, and study experimentally how radio interference affects the available MAC protocols in Contiki, namely X-MAC [49], low-power probing [144], and CoReDac [214], as well as TinyOS's low-power listening (LPL) [164].

X-MAC is a power-saving MAC protocol in which senders use a sequence of short preambles (strobes) to wake up receivers. Nodes turn off the radio for most of the time to reduce idle listening and wake up shortly at regular intervals to listen for strobes. When a receiving node wakes up and receives a strobe destined to it, it replies with an acknowledgement indicating that it is awake. After receiving the ACK, the sender transmits the data packet, as shown in Figure 4.9(a). The X-MAC implementation in Contiki has several parameters of significance to our experiments. The `ontime` determines the maximum time that a receiver listens for strobes, whereas `offtime` specifies the time to sleep between waking up to listen for strobes. The `strobe_time` denotes the duration a sender transmits strobes until it receives a strobe acknowledgement from the receiver. In the default Contiki X-MAC implementation, `strobe_time` = `offtime` + (20 × `ontime`).

We consider a Low-Power Listening (LPL) layer that implements an asynchronous wake-up scheme for CC2420 radios. Nodes periodically wake up to detect transmissions by relying on CCA. Unlike X-MAC, senders repeatedly transmit the entire packet for twice the duration of the wake-up period. In case of unicast transmissions, the intended receiver may acknowledge the transmission to notify the sender on correct packet delivery so that the sender can stop transmitting earlier. To implement this functionality, packet transmissions are interleaved with periods of silence in order to allow ACK transmissions. The only LPL parameter tunable by the users is the wake-up period.

Low-Power Probing (LPP) is a power-saving MAC protocol where receivers periodically send probes to announce that they are awake and ready to receive a data packet and keeps their radio on for a short time to listen for data packets [144]. A node willing to send a packet turns on its radio waiting for a probe from a neighbour it wants to send to. On the reception of a probe from a potential receiver, the node sends an ACK before the data packet, as shown in Figure 4.9(b). The LPP implementation in Contiki contains two important parameters. The `ontime` determines how long a receiver keeps the radio on



(a) Packet reception rate

(b) Power consumption (transmitter)

Figure 4.10: Performance of MAC protocols under semi-periodic interference [43].

after the transmission of a probe, whereas the `offtime` is the time between probes. We use $\frac{1}{2}$ and $\frac{1}{64}$ seconds for `offtime` and `ontime` respectively. Another parameter is the time to keep an unsent packet: Contiki LPP's default value is $4\times(\texttt{ontime}+\texttt{offtime})$. If LPP receives a packet from the network layer when the packet queue is full, it discards the new packet. The queue length is configurable, and the default size is 8 packets.

CoReDac is a TDMA-based convergecast protocol [214] that builds a collection tree that guarantees collision-free radio traffic. To avoid collisions among packets from their children, CoReDac parents split their reception slots into sub-slots, and assign one to each child. Packet acknowledgements are pivotal in CoReDac because they piggyback the assignment information, and they are used for synchronizing the TDMA-schedules. A node that misses an acknowledgement must keep its radio on until it hears a new one.

Experimental setup. We carry out experiments with several pairs consisting of a sender node $\mathcal{S}$ and a receiver node $\mathcal{R}$. Node $\mathcal{S}$ periodically transmits unicast packets with a payload of 22 bytes to $\mathcal{R}$, with a period uniformly distributed between 0.75 and 1.25 s. For each pair, we place one HandyMote $\mathcal{H}$ between $\mathcal{S}$ and $\mathcal{R}$, and make sure that $\mathcal{H}$'s transmission power is sufficiently high to block the communication between the two nodes. Each HandyMote reproduces two distinct interference patterns: (i) continuous blocks of interference with uniformly distributed duration and spacing (*bursty interference*), and (ii) continuous blocks of interference, but with a significantly smaller variance in the duration of the burst and their period (*semi-periodic interference*)[5]. In our experiments, we use two metrics to measure the performance of a protocol: packet reception rate (PRR) and power consumption. The latter is computed at the sender[6] using Contiki's software-based power profiler [73] (the same mechanism was implemented in TinyOS for LPL's experiments).

[5] Full details about the experimental setup and the comparison between the performance of state-of-the-art MAC protocol can be found in Paper A included in this thesis [43].

[6] In the remainder of this section, we refer to power consumption as the average power consumed by the transmitter's radio in receive mode (RX power) during the full experiment. We focus on the transmitter's RX power for two main reasons: first, all the selected protocols shift the burden to the transmitter. Second, in all our experiments the RX power is at least an order of magnitude larger than the transmit power (TX power).

Semi-periodic interference. We first investigate the performance of protocols in the presence of semi-periodic interference. We vary the rate of interference (i.e., the percentage of time during which the channel is busy) generated by the HandyMotes between 0 and 60, and collect several thousands packets for each measurement.

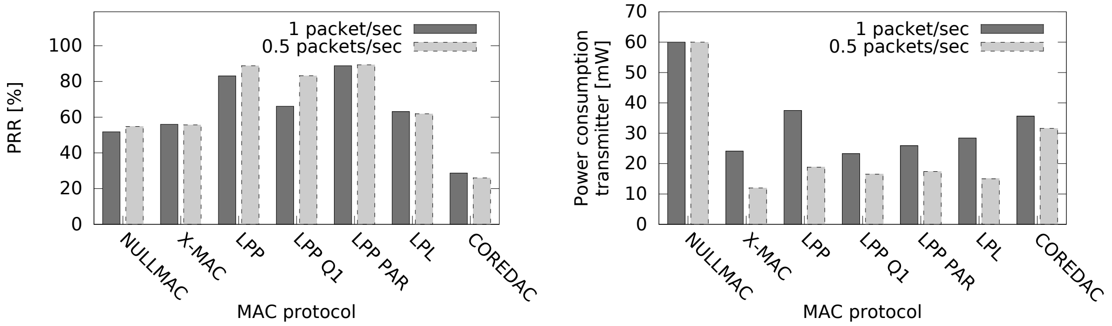
Figure 4.10 shows the results of our experiments. To begin, we observe the performance of nullMAC, a simple MAC layer that just forwards packets between the radio driver and the network layer without any duty-cycling mechanism. We use nullMAC to verify the correctness of our setup: the PRR decreases linearly with the interference rate, and the average power consumption is 60 mW independently on the interference pattern (the radio is indeed always on, causing a power consumption of approximately 20 mA (the sensor nodes are powered with 3 V).

We then investigate three different variants of low-power probing: the default version (LPP), a version called LPP-Q1 that does not employ a queue (i.e., a new packet from the upper layer is discarded in case the previous one has not yet been transmitted by the MAC), and a version called LPP-PAR in which the receiver transmits a new probe immediately after a packet reception. Figure 4.10(a) shows that all variants of LPP have fairly high packet reception rates compared to the other protocols we consider. Among LPP-based solutions, the best performance is obtained with LPP-PAR, where the receiver transmits a new probe immediately after a packet reception. By doing so, the sender can drain its queue when the interference clears and sustain a high PRR also under high interference by deferring transmissions until interference is over. Figure 4.10(b) shows that the power consumption of LPP-Q1 is substantially lower than the default version of LPP. The reason for this is the lower PRR shown by LPP-Q1: with fewer packets to be transmitted, the radio is turned off more often. This difference becomes very apparent at an interference rate of 60%, where LPP has its radio turned on almost all the time since there is almost always a packet in the queue waiting to be transmitted. In contrast to the default LPP, LPP-PAR can quickly drain its queue during interference-free periods and hence turn off quickly its radio, saving a substantial amount of power.

X-MAC sustains a PRR similar to (but slightly higher than) nullMAC: in X-MAC, the sender's `strobe_time` takes a little longer than the receiver's `offtime`, and the receiver has hence on average more than one chance to hear a strobe. The same reasoning should also apply for CoReDac. However, at higher interference, a higher packet loss leads to a desynchronization, and the performance of this TDMA-based protocol drastically degrades.

Nodes employing LPL are reasonably effective at detecting the presence of interference when the interference rate is lower than 60%, and packet losses occur mostly because of data corruption. On the other hand, at 60% interference it is often the case that the CCA mechanism never finds the channel free. After a maximum number of tries, the packet is dropped on the sender side, causing a drastic decrease in PRR. The increasing power consumption shown for LPL in Figure 4.10(b) is simply an effect of the decreasing PRR: the fewer packets are received, the less likely is the sender to receive the ACK and to terminate the transmissions earlier.

Bursty Interference. We carry out the same set of experiments by letting the HandyMotes reproduce bursty interference, and investigate how performance changes depending on the network load. Figure 4.11 illustrates the results: for most MAC protocols the PRR does not change depending on the transmission rate. In most cases, indeed, the interfe-



(a) Packet reception rate

(b) Power consumption (transmitter)

Figure 4.11: Performance of MAC protocols in the presence of bursty interference [43] using different transmission rates.

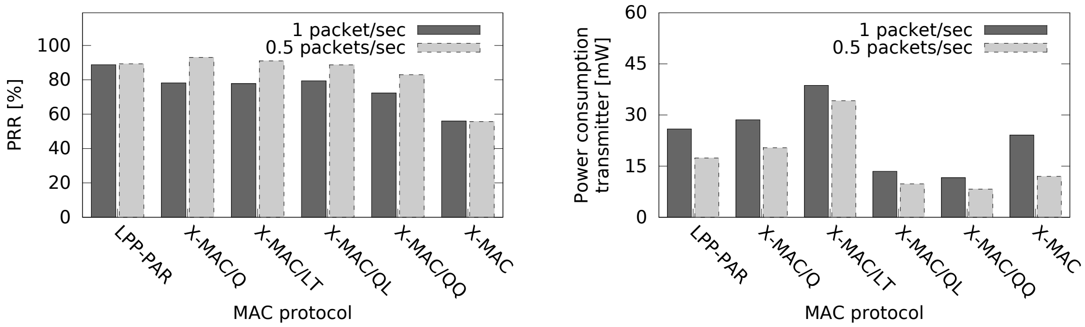
rence rate is what ultimately determines the observed PRR. An exception is LPP-Q1, where the PRR increases by almost 10% when the application transmits packets less frequently. The reason is that with higher transmission rates, a packet cannot be sent before the application hands the next packet to the MAC layer, and thus the latter packet is discarded. This can either happen with long periods of interference, or when periods of interference overlap with the instants in which the receiver sends probes.

4.3.2 Improving X-MAC's robustness to interference

The results presented in Section 4.3.1 hint that, to be robust to interference, existing MAC protocols should (i) hold a packet longer so that multiple handshake attempts are possible, (ii) implement packet trains as a means to quickly send multiple packets that have accumulated during bursts of interference, and (iii) apply suitable congestion CCA backoff schemes. Starting from Contiki's X-MAC implementation, we now develop different versions of the protocol that embed these mechanisms. We then evaluate their reliability and measure their energy-efficiency.

Longer strobing intervals (X-MAC/LT). We first create X-MAC/LT, a protocol identical to X-MAC, except for one parameter, strobe_time, which we increase from (offtime + 20 × ontime) to (4 × offtime + 20 × ontime). This allows X-MAC/LT to hold packets longer: we expect this change to lead to a significantly higher PRR, at the cost of a higher energy expenditure.

Packet queue with fast drain (X-MAC/Q). A second enhanced version of X-MAC contains a packet queue implemented by using a statically allocated array of packets and their corresponding attributes. By default, the queue stores up to four unicast packets: the latter are not sent directly, but instead linger shortly for a configurable time ($\frac{1}{32}$ s in our experiments). The linger time makes it possible to accumulate packets into the queue, which allows the layer on top of X-MAC to create a burst of packets. When the accumulation timer has expired, X-MAC/Q gets the oldest packet from the queue, and immediately starts sending strobes to the addressed receiver. To enable fast queue draining, each strobe contains the number of packets for the destination that the sender



(a) Packet reception rate

(b) Power consumption (transmitter)

Figure 4.12: Reliability and energy-efficiency of the enhanced X-MACs using different transmission rates [43].

has in its queue. If the sender receives a strobe ACK within a configured waiting time, it sends one packet at a time, including the strobe procedure, separated by a very short time ($\frac{1}{128}$ s) instead of the usual duty-cycle interval. If the sender does not receive the strobe acknowledgement, a new attempt comes after $\frac{1}{32}$ s. Packets are removed from the queue when they have either been successfully sent, or timed out after ten seconds.

Aggressive congestion back-off (X-MAC/QL and X-MAC/QQ). We further extend X-MAC/Q to include a linear and a quadratic CCA congestion back-off timer and call the two versions X-MAC/QL and X-MAC/QQ, respectively. The protocols use a single CCA check to verify whether the channel is clear before sending out the first strobe. If the CCA check fails, X-MAC/QL and X-MAC/QQ wait respectively for ($\frac{1}{128} \times \texttt{number\_of\_attempts}$) and ($\frac{1}{128} \times \texttt{number\_of\_attempts}^2$) milliseconds before the next attempt.

Experimental evaluation. We now compare the performance of our enhanced X-MAC versions in the presence of interference using the same experimental setup described in Section 4.3.1 (bursty interference). Figure 4.12(a) shows that both X-MAC/Q and X-MAC/LT significantly increase the packet reception rate compared to the default X-MAC (we also include LPP-PAR in the figures to have an additional comparison). When each node transmits one packet every two seconds, X-MAC/LT sustain up to 20% more PRR with respect to the default X-MAC.

Although X-MAC/Q and X-MAC/LT show a similar PRR, the power consumption is much higher for X-MAC/LT than for X-MAC/Q, as expected (Figure 4.12(b)). The energy efficiency increases significantly when using aggressive congestion back-off times: the energy consumption of X-MAC/QQ and X-MAC/QL is essentially halved compared to X-MAC/Q, without affecting the overall PRR significantly. Compared to X-MAC/QQ, X-MAC/QL consumes slightly more energy but achieves a higher PRR: on the one hand, the linear backoff causes more frequent samples of the channel than the quadratic one does, leading to higher power consumption; on the other hand, the quadratic algorithm may grow its sampling interval exponentially up to a point where expired packets will be removed from the queue.

In summary, our experimental evaluation shows that the enhanced versions of X-MAC are actually more robust to interference and sustain a higher PRR than the original implementation. To achieve a good energy-efficiency, the use of congestion back-off times seems highly beneficial: X-MAC/QQ and X-MAC/QL's power consumption is even lower than X-MAC's despite that they achieve a much higher PRR.

4.4 JAG: Reliable Agreement Despite Interference

In Section 2.3.3 we have highlighted how agreeing on fundamental pieces of information (e.g., a TDMA schedule) represents a challenge in congested channels, where wireless sensor nodes have very low chances to successfully deliver a packet. We now focus on the unicast case (agreement between two neighbouring nodes) and analyse the limitations of traditional message-based approaches in the presence of radio interference, showing their inefficiency even when using short ACK packets (Section 4.4.1). We therefore propose to use short handshakes, and to replace the last ACK packet with a jamming sequence, as this can be used to reliably inform about the correct reception of a message carrying the information to be agreed upon (Section 4.4.2). Based on this insight, we design and implement JAG[7], an agreement protocol for WSNs exposed to external interference (Section 4.4.3). We illustrate how JAG can be parametrized to obtain predictable performance (Section 4.4.4) and show that it outperforms traditional packet-based agreement protocols in the presence of interference with respect to agreement probability, energy consumption, and time-to-completion (Section 4.4.5).

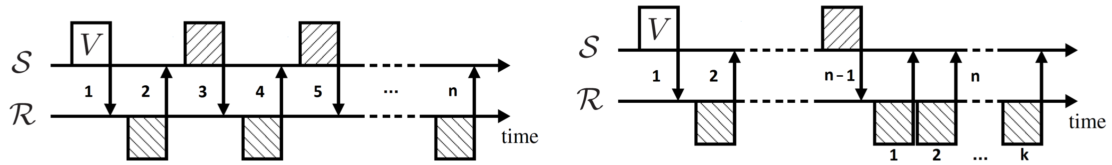
4.4.1 The two general's agreement problem

Agreeing on a given piece of information is a classical coordination problem in distributed computing. The two generals' agreement problem, formulated by Jim Gray to illustrate the two-phase commit protocol in distributed database systems [90], is often used to explain the challenges when attempting to coordinate an action by communicating over a faulty channel, and can be described as follows.

Two battalions are encamped near a city, ready to launch the final attack. Because of the redoubtable fortifications, the attack must be carried out by both battalions at the same time in order to succeed. Hence, the generals of the two armies need to agree on the time of the attack, and their only way to communicate is to send messengers through the valley. The latter is occupied by the city's defenders, and a messenger can be captured and its message lost, i.e., the communication channel is unreliable. Since each general must be aware that the other general has agreed on the attack plan, messengers are used also to exchange acknowledgements. However, because the acknowledgement of a message receipt can be lost as easily as the original message, a potentially infinite series of messages is required to reach an agreement.

In the context of wireless communications, the problem can be rephrased as follows. When two nodes, $\mathcal{S}$ and $\mathcal{R}$, need to agree on a common value V, they exchange a sequence of n messages in an alternating manner (Figure 4.13(a)). Node $\mathcal{S}$ is the initiator of the exchange. After the transmission of V, each subsequent message acknowledges the receipt

[7] Full details about JAG's design and implementation can be found in Paper D included in this thesis [46].



(a) n-way handshake between nodes $\mathcal{S}$ and $\mathcal{R}$ (b) Enhanced n-way handshake with redundancy

Figure 4.13: n-way handshake between nodes $\mathcal{S}$ and $\mathcal{R}$ (a) and an enhanced n-way handshake using redundancy: the last ACK message is transmitted k times [46].

of the previous message, i.e., a node sends message $i > 1$ only if it correctly received message $i-1$. Each node uses a simple rule to determine the success of the exchange: if all expected messages are received, the exchange is deemed successful, otherwise the exchange is deemed unsuccessful.

This scenario corresponds to an *n-way handshake* between $\mathcal{S}$ and $\mathcal{R}$, where n is the number of packets exchanged. The n-way handshake is a widely used mechanism in communication networks: TCP employs a 3-way handshake to establish connections over the network, whereas IEEE 802.11i uses a 4-way handshake to carry out the key exchange.

An n-way handshake can have three possible outcomes:

1. **Positive Agreement.** The n messages are all received correctly, and both nodes deem the exchange as successful, accepting V.

2. **Negative Agreement.** A message m with $m < n$, i.e., a message prior to the last message n, is lost. None of the nodes receives all the expected messages, hence both nodes deem the exchange as unsuccessful, discarding V.

3. **Disagreement.** The last message is lost. One of the nodes receives all the expected messages, deems the exchange as successful and accepts V; whereas the second node misses the last message and deems the exchange as unsuccessful, rejecting V.

In the original two generals' scenario, a *positive agreement* would lead to a simultaneous attack of the city by both battalions and a consequent victory, a *negative agreement* would cause both battalions to stall, while a *disagreement* would trigger the attack of only one battalion and a consequent defeat of the attacking forces.

While *disagreements* are potentially fatal, *negative agreements* are often less severe. For example, if the shared value contains the next channel to be used for communication, two nodes are better off staying in the same lossy channel, rather than having only one of them move to a different frequency. The probability of negative agreements should, however, be minimized, as it may lead to reduced performance.

Hence, an agreement protocol should strive to minimize disagreements as a first priority, maximize positive agreements as a second (almost equally high) priority, and minimize negative agreements as a third (lower) priority. A metric to measure the quality of an agreement protocol (whose value should be minimized) is therefore the *DPA ratio*, where:

$$DPA\ ratio = \frac{\text{Prob}(Disagreement)}{\text{Prob}(PositiveAgreement)} \tag{4.7}$$

The importance of the last message. In an n-way handshake, disagreements only occur if the *last* message is lost. Hence, one should attempt to increase the probability of successfully delivering the last packet. A possibility is to use redundant packet transmissions (i.e., repeating a message several times and assuming successful transmission if at least one copy is received). For example, one could devote extra-resources and use an enhanced n-way handshake in which the last packet is repeated k times, as shown in Figure 4.13(b).

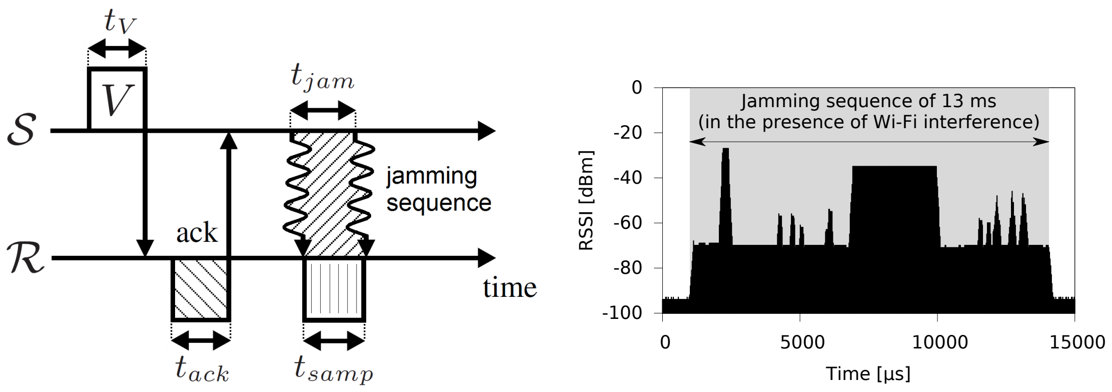
Another possibility is to use short messages and send them as close as possible to each other, in order to increase the chances of fitting the whole handshake into an idle period. In principle, the longer the idle period and the shorter the handshake, the higher the likelihood of obtaining positive agreements. The ability of modern IEEE 802.15.4-compliant radios to automatically generate and send ACKs for data frames in hardware [167] can be very helpful to minimize the duration of a handshake. However, hardware ACKs cannot be used to carry out a complete n-way handshake (with $n > 2$), since they cannot be used in reply to another hardware ACK. Imagine a node $\mathcal{S}$ starting a handshake by sending a message to $\mathcal{R}$. The latter can reply with a hardware ACK, but $\mathcal{S}$ will have to receive and extract the packet, analyse its validity, as well as to prepare a new ACK frame, load it into the buffer, and send it over-the-air (in case a train of k redundant software ACKs is sent, the packet can be loaded into the buffer once and sent repeatedly). This may cause long latencies that break the agreement in the presence of the short idle periods commonly generated by Wi-Fi transmissions.

Furthermore, it is also highly inefficient to encode the binary information carried by an ACK message inside an IEEE 802.15.4 frame, especially in the presence of interference. Despite the payload contains only a single ACK bit, the whole packet consists of a synchronization preamble, a physical header (6 bytes), as well as a MAC header and footer (from 9 to 29 bytes): if any of the bits in the headers and preamble is corrupted by interference, the packet may not be correctly decoded [107, 132].

4.4.2 Jamming as binary ACK signal

Therefore, instead of encoding the last ACK as packet transmission, we propose to encode it by means of *jamming*, where the presence of a jamming sequence signals the receipt of the previous message. The key advantage of this approach is that precisely timed jamming signals can be reliably detected on off-the-shelf motes even under heavy interference by means of high-frequency RSSI sampling. In the presence of additional external interference, the RSSI register will return the maximum of the jamming signal and the interference signal due to the co-channel rejection properties of the radio [42]. Figure 4.14(b) illustrates this for a jamming signal sent in the presence of Wi-Fi interference. As we have shown in Section 4.1.2, typical interference sources – in contrast to a jamming signal – do not produce continuous interference "spikes" for long periods of time, rather they alternate between short idle and busy periods. That is, if the jamming signal lasts longer than the longest busy period of the interference signal, we are unequivocally able to detect the absence of the jamming signal by checking if any of the RSSI samples equals the sensitivity threshold of the radio.

Identification of the interfering source. While a jamming signal can encode the binary acknowledgement information, it cannot encode the identities of sender and receiver



(a) JAG's three-way handshake

(b) Distinguishing a jamming signal from interference using high-frequency RSSI sampling

Figure 4.14: JAG's working principle. The last acknowledgement of a 3-way handshake is sent in the form of a jamming signal (a). The latter can be reliably distinguished from other interference sources by means of high frequency RSSI sampling (b) [46].

as a regular packet would. When carrying out a handshake, however, these identities are already included in the message V to be acknowledged, and therefore are implicitly known to the two nodes, as long as the communication channel remains allocated exclusively for the whole duration of an exchange (i.e., intra-network interference is avoided). Any protocol that embeds JAG as a building block for agreement needs to meet this requirement. At the MAC layer, RTS/CTS can be used to allocate the channel in CSMA protocols, whereas in TDMA protocols the timeslot must be long enough to complete an exchange.

4.4.3 Protocol design

JAG (Jamming-based AGreement) employs a three-way handshake in which the last ACK is sent in the form of a jamming signal as shown in Figure 4.14(a). The choice of three-way handshakes (as opposed to two-way) is motivated by two facts. First, a three-way handshake increases the reliability of identifying the jamming signal, as it provides a reference RSSI value. Second, three-way handshakes avoid disagreements due to asymmetric links: for instance, if $\mathcal{S}$ has a link with $\mathcal{R}$ but the reverse link is not present, a two-way handshake would always lead to disagreements, since $\mathcal{R}$ is not able to confirm the reception of V.

The protocol proceeds as follows. $\mathcal{S}$ initiates the exchange and sends the information V towards a receiver $\mathcal{R}$. If V is successfully received, $\mathcal{R}$ saves the signal strength r_s of the received packet and sends an ACK message back to $\mathcal{S}$. We can send either hardware or software acknowledgements: in the remainder of this paper we assume that hardware ACKs are available. If $\mathcal{S}$ receives the acknowledgement, it transmits a jamming signal for a period t_{jam}. Meanwhile, $\mathcal{R}$ carries out a high-frequency RSSI sampling for a period $t_{samp} \leq t_{jam}$ that is synchronized in such a way that the fast RSSI sampling is carried out while the jamming signal is on the air. The message V is used as the synchronization signal: given that clock drift is not too high at time-scales of a few milliseconds, it is sufficient to include a short safety margin to compensate for drift. For simplicity, in the rest of the paper, we assume $t_{jam} = t_{samp}$. If $\mathcal{R}$ detects the presence of the jamming signal, it deems the exchange as successful; otherwise, V is discarded. $\mathcal{S}$ deems the exchange as successful if the ACK is received within a short time-out period, otherwise the jamming sequence is not generated and the handshake immediately terminated.

After the reception of V, node $\mathcal{R}$ carries out a high-frequency RSSI sampling to detect the absence or the presence of the jamming sequence transmitted by $\mathcal{S}$. The method to detect the jamming signal is simple: if a jamming sequence is sent, *all* RSSI samples should be above r_{noise}, with the latter being the RSSI noise floor threshold of the radio. Hence, if, during t_{samp}, $\mathcal{R}$ observes *at least one RSSI sample* with a value comparable to r_{noise}, it concludes that the jamming sequence was not transmitted.

This process can be described as follows. Denoting $\{x_1, x_2, \ldots, x_n\}$ as the sequence of RSSI values sampled during t_{samp}, we define the binary sequence $\{X_1, X_2, \ldots, X_n\}$ as follows: if $x_i \leq r_{noise}$, then $X_i = 1$, else $X_i = 0$. $\mathcal{R}$ makes a decision about the presence of the jamming sequence as follows: if $\sum_{i=1}^{n} X_i = 0$, then $\mathcal{S}$ was transmitting a jamming signal and hence V is accepted; otherwise, V is discarded.

Using this algorithm, JAG would operate correctly and would be able to recognize the presence or absence of a jamming signal reliably. However, we can enhance its performance by exploiting the knowledge of the received signal strength r_s of the packet containing the information V.

The role of r_s. Under the hypothesis that the jamming signal has a reasonably similar signal strength to r_s (RSSI does not change significantly between consecutive transmissions spaced by only a few milliseconds), $\mathcal{R}$ can filter out any interference source weaker (i.e., resulting in an RSSI range smaller) than $(r_s - \Delta_r)$, with Δ_r being a tolerance margin to compensate for the inaccuracy of low-power radios and the instability of the RSSI readings. This allows to shorten t_{jam} and achieve a higher energy efficiency: the higher the configurable threshold R_{thr}, the shorter the duration of busy periods.

Hence, if $(r_s - \Delta_r) > r_{noise}$, JAG 's algorithm is executed as follows: if $x_i < (r_s - \Delta_r)$, then $X_i = 1$, else $X_i = 0$. $\mathcal{R}$ still makes a decision about the presence of the jamming sequence in the following way: if $\sum_{i=1}^{n} X_i = 0$, then $\mathcal{S}$ was jamming and hence V is accepted; otherwise, V is discarded.

Furthermore, r_s also increases the reliability of fast RSSI sampling. The maximum distance over which a packet can be successfully received and decoded is shorter than the distance over which a jamming signal can be captured. This may lead to confusion in a scenario in which two nodes that cannot communicate with each other are allocated the same time slot in a TDMA protocol and transmit a message concurrently. By using a threshold r_s, we make sure that a receiver $\mathcal{R}$ is in the communication range of $\mathcal{S}$, and therefore r_s cannot be achieved by any other node transmitting simultaneously.

JAG implementation. We implement JAG on Maxfor MTM-CM5000MSP and Sentilla Tmote Sky nodes. Our implementation, based on Contiki [72], uses two main building blocks: the high-frequency RF noise measurement presented in Section 4.1.1, and the generation of a jamming sequence. The latter follows the approach used in JamLab to generate precisely timed jamming signals: by configuring the `MDMCTRL1` register, the CC2420 radio can output a continuous modulated carrier signal that is stable over time [35, 39, 42]. This approach is superior to packet-based jamming, as the generated signal is independent of both packet sizes and inter-packet times.

For all our experiments we use Contiki's nullMAC and nullRDC to avoid protocol-specific implementations: the MAC layer hence just forwards packets to the upper or lower protocol layer and does not perform any duty cycling. To ensure that the execution time of the entire handshake is bounded and independent of CCA back-off times, we do not postpone transmissions until the channel becomes clear. Instead, we carry out a single clear channel assessment before sending V: if the channel is found busy, the transmission is cancelled. This is an optimization, as sending V despite the busy channel would result in a negative agreement (V would be lost).

To ensure alignment between jamming t_{jam} and sampling t_{samp}, we implement a simple synchronization mechanism. $\mathcal{S}$ and $\mathcal{R}$ synchronize their operations based on the reception of V: the transmission or reception of the SFD is used as the synchronization signal.

4.4.4 Predictable performance

The length of the jamming sequence t_{jam} can be tuned in order to provide probabilistic guarantees on the fraction of disagreements. Denoting t_{busy}^{max} as the maximum busy period that can be encountered in the presence of interference, we can guarantee that $\mathcal{S}$ and $\mathcal{R}$ will agree on V by setting $t_{jam} > t_{busy}^{max}$. In such a case, an idle period will surely be encountered during t_{samp}, and the absence of a jamming sequence unequivocally detected. Hence, the most pernicious outcomes (disagreements) are eliminated, and only positive or negative agreements can occur.

In some scenarios, however, one may need to know the outcome of the agreement process before t_{busy}^{max}. In these cases, where $t_{jam} \leq t_{busy}^{max}$, disagreements may still occur. For this type of scenarios, given t_{jam}, we derive an upper bound for the probability of obtaining disagreements. In this way, a user with stringent real-time constraints can assess if the fraction of disagreements is within the limits permitted by the QoS requirements of the application.

Probabilistic model. To bound the probabilities of positive agreements and disagreements given a certain value of t_{jam}, we now derive a probabilistic model that can be parametrized by means of high-frequency RF noise measurements. The parametrization is typically carried out before the actual deployment, but it would also be possible to characterize interference at runtime, for example in case the RF environment has changed significantly from the prior observation. The user needs to follow three simple steps: (i) compute the *pdf* of idle periods $p(i)$, where i represents the length of the idle period (Section 4.1.3), (ii) compute the conditional *pdf* of the busy periods following the idle periods $p(b > x|i)$, and (iii) use this probabilistic model to obtain the value of t_{jam} that provides the desired QoS.

In order to derive the probabilities of positive agreements and disagreements, we need to understand the interplay between the length of an idle period i and the 3-way handshake method used by JAG (i.e., the transmission of the packet embedding V, the ACK, and the jamming signal). In JAG, if $\mathcal{R}$ sends the ACK, four outcomes can occur: (i) a positive agreement, if the ACK is successfully delivered to $\mathcal{S}$ and the jamming signal is correctly decoded by $\mathcal{R}$; (ii) a negative agreement, if the ACK is lost and $\mathcal{R}$ detects the *lack of* jamming; (iii) *another positive agreement*, independently of the fact that the ACK is received or not if, after sending the ACK, $\mathcal{R}$ detects an interference signal with a strength higher than the expected jamming signal and hence assumes a successful transaction ($\mathcal{R}$

is assuming that the jamming signal was buried within the stronger signal); and (iv) a disagreement, if the ACK is lost, but, by chance, a high interference signal lasts longer than t_{samp}. In this case, $\mathcal{R}$ assumes, mistakenly, a successful exchange, i.e., a negative agreement turns into a disagreement.

Hence, in JAG, positive agreements are given by the following equation:

$$P_{\text{jam}}\{\text{Pos. Agr.}\} = \sum_{i>t_{\text{pkt}}+t_{\text{ack}}}^{\infty} s(i)(1 - \frac{t_{\text{pkt}} + t_{\text{ack}}}{i}) \tag{4.8}$$

whereby the first term of the product states the probability of obtaining an idle slot of length i, and is computed using Equation 4.2. The second term of the product states the probability that the selected idle slot can “contain” the transmission of the packet (t_{pkt}) followed by the transmission of the ACK (t_{ack}).

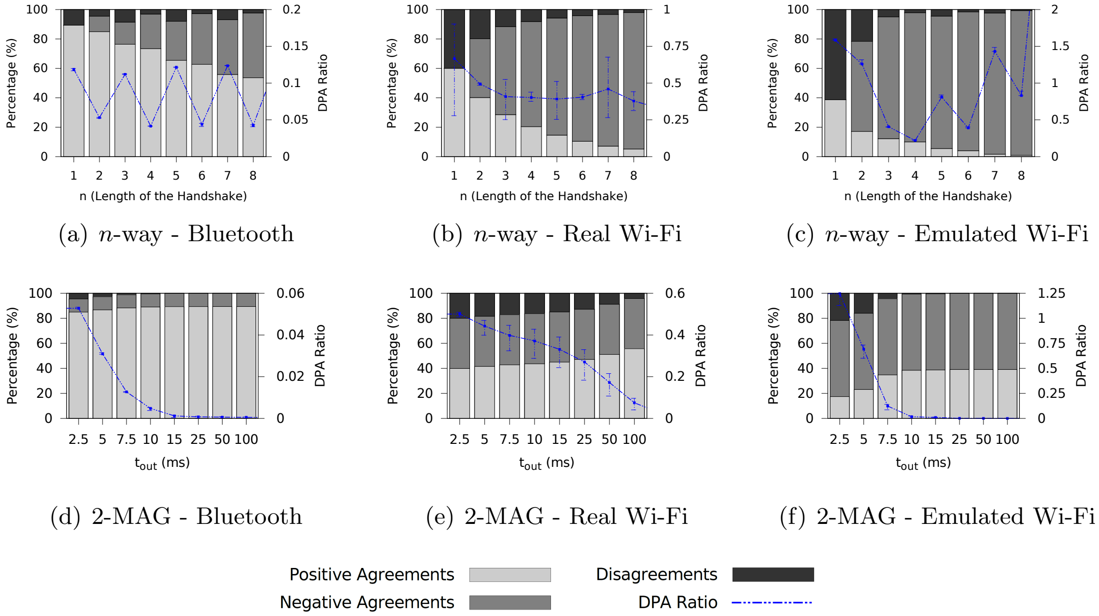
In order to obtain the fraction of disagreements, we use a bounding probability. There are three necessary but not sufficient conditions to obtain disagreements: (i) the packet embedding V is transmitted successfully, (ii) the ACK is corrupted and (iii) the interference signal after the ACK is longer than t_{jam} (to shadow the jamming signal). Hence, we define the probability of obtaining disagreements with JAG as follows:

$$\begin{aligned} &P_{\text{jam}}\{\text{Disagreement}\} \leq \sum_{i=1}^{t_{ack}} s(i)p(b > t_{jam}|i) \cdot 1 + \\ &\sum_{i>t_{ack}}^{t_{pkt}+t_{ack}} s(i)p(b > t_{jam}|i)(1 - \frac{\min(t_{pkt}, i)}{i}) + \sum_{i>t_{pkt}+t_{ack}}^{\infty} s(i)p(b > t_{jam}|i)(\frac{t_{ack}}{i}) \end{aligned} \tag{4.9}$$

Each of the sums on the right side of the equation has three terms. The first term $s(i)$ denotes the probability of obtaining an idle slot of length i (Equation 4.2). The second term $p(b > t_{jam}|i)$ denotes the probability of obtaining a busy period b longer than t_{jam} after an idle period of length i (the minimum requirement to shadow the jamming signal). The third term differs for each sum, and denotes the probability that the ACK will be corrupted: in the first summation this probability is 1, because the idle time is less than t_{ack}, i.e., the ACK will always be corrupted; in the second and third summations, this probability describes the chances that the agreement starts early enough to allow a successful delivery of PKT, but late enough to corrupt the ACK. Please note that, in Equation 4.9, the term $p(b > t_{jam}|i)$ assumes that the corrupted ACK ends exactly before the next busy period starts. In practice, the ACK will likely have a Δ overlap with the beginning of the busy period b, and hence, b will need to be longer than $(t_{jam} + \Delta)$ to lead to a disagreement. Given that $p(b > t_{jam}|i) > p(b > (t_{jam} + \Delta)|i)$, in practice, we can expect a lower fraction of disagreements.

4.4.5 Evaluation

We use our local university testbed with 15 Maxfor MTM-CM5000MSP nodes to evaluate the performance of several agreement protocols under different types of interference. To this end, we use both JamLab to emulate the interference produced by Bluetooth and Wi-Fi devices, as well as a laptop continuously downloading a file from a nearby Wi-Fi access point. We validate our first set of results using a second testbed deployed in residential



(a) n-way - Bluetooth

(b) n-way - Real Wi-Fi

(c) n-way - Emulated Wi-Fi

(d) 2-MAG - Bluetooth

(e) 2-MAG - Real Wi-Fi

(f) 2-MAG - Emulated Wi-Fi

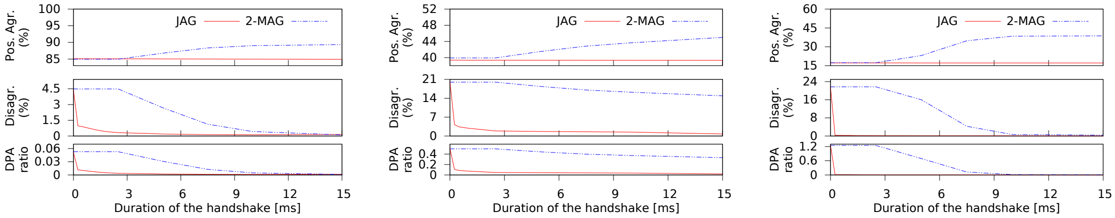
Figure 4.15: Performance of a packet-based n-way handshake and of 2-MAG (an enhanced 2-way handshake in which the ACK is sent repeatedly multiple times) under different types of interference [46]. In 2-MAG, it is sufficient to receive one ACK packet within a time t_{out} to consider the exchange successful, hence the longer t_{out}, the better 2-MAG will perform (at the price of an increased energy consumption).

buildings surrounded by Wi-Fi stations: we run different agreement protocols for several days and compare their performance over time.

In all our experiments, we use several pairs of nodes $\mathcal{S}$ and $\mathcal{R}$. Node $\mathcal{S}$ always initiates the handshake, and transmits a data packet composed of a 6-byte payload containing the information to be agreed upon V and the transmission power used T_P. For each handshake (which is initiated after a random interval in the order of hundreds of milliseconds), we select a random transmission power between -25 dBm and 0 dBm in order to create different types of links. $\mathcal{R}$ replies to the packet using T_P, i.e., the same transmission power used by $\mathcal{S}$. Hardware ACKs are enabled by default, and nodes remain on the same channel during the whole duration of the experiment, in which we perform several hundred thousand handshakes.

Packet-based n-way handshake. We firstly analyse the performance of the packet-based n-way handshake shown in Figure 4.13(a) (redundancy factor $k = 1$) under different interference patterns. In our implementation, every packet from $\mathcal{R}$ to $\mathcal{S}$ is sent using the hardware ACK support, so to minimize the latency between the reception of the previous packet and the dispatch of the following one.

Figures 4.15(a), 4.15(b), and 4.15(c) show the percentage of positive/negative agreements and disagreements obtained under different interference patterns. The values are computed as an average over all transmission power values T_P used in our experiments, excluding the ones leading to asymmetric links. We can observe that, regardless of the



(a) Bluetooth (b) Real Wi-Fi (c) Emulated Wi-Fi

Figure 4.16: Compared to 2-MAG (2-way handshake in which the last ACK packet is sent k times), JAG sustains significantly less disagreements (although also less positive agreements) independently of the interfering source [46].

interference source, the longer the handshake, the smaller the number of disagreements and positive agreements. Hence, the DPA ratio does not decrease when increasing the length of the handshake n, hinting that packet-based n-way handshakes are not optimal under external interference.

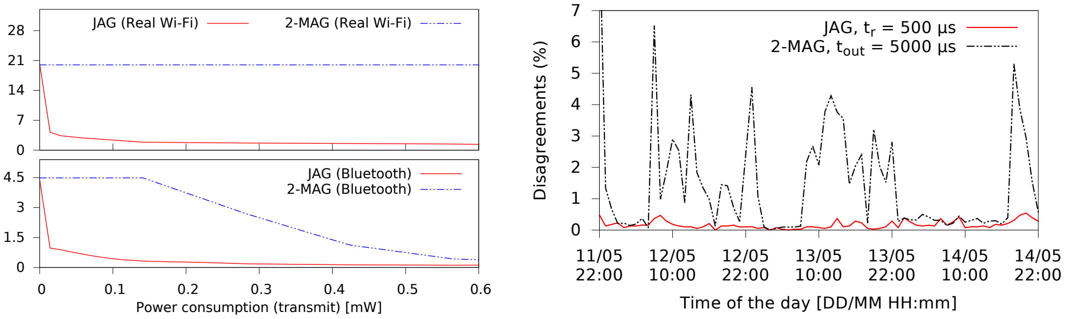
2-way handshake enhanced with redundancy (2-MAG). To minimize the DPA ratio, we introduce redundancy of the last ACK as discussed in Section 4.4.1, and analyse the performance of a 2-way handshake in which the last ACK packet is sent k times, as illustrated in Figure 4.13(b) (we refer to this protocol as 2-MAG – 2-way handshake Message-based AGreement). The choice of a 2-way handshake is driven by the results obtained above: a low n minimizes the probability of negative agreements, and therefore there are higher chances that 2-MAG sustains more positive agreements thanks to its redundant transmissions. We make sure to carry out a fair comparison by eliminating asymmetric links that would always lead to disagreements when using a 2-way handshake.

In our implementation, hardware ACKs are enabled, i.e., the first ACK sent from $\mathcal{R}$ to $\mathcal{S}$ has a short and fixed-delay latency. Every other ACK will be generated via software by pre-loading the ACK into the radio buffer and by repeatedly sending its content k times. Please note that the preparation of the software ACK is time-critical, as one needs to extract and analyse V before creating and loading the ACK into the radio buffer.

In order for $\mathcal{S}$ to consider V as successfully exchanged, it is sufficient to receive one ACK packet within a maximum waiting time t_{out}. Clearly, the longer t_{out}, the higher the likelihood that at least one ACK packet will be correctly decoded and the better 2-MAG will perform (at the price of an increased energy consumption). Hence, we compute t_{out} as the maximum time in which node $\mathcal{S}$ waits for a valid ACK packet from $\mathcal{R}$.

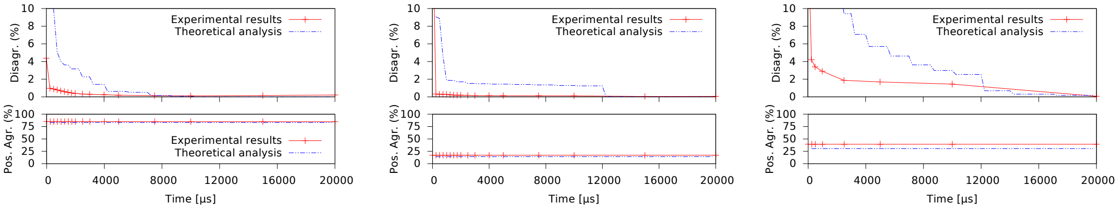
Figures 4.15(d), 4.15(e), and 4.15(f) show the percentage of positive and negative agreements as well as disagreements obtained in the presence of interference using 2-MAG as a function of t_{out}. As expected, the longer t_{out}, the lower the number of disagreements in favour of positive agreements. As this minimizes the DPA ratio, 2-MAG outperforms a generic n-way handshake without redundancy in the presence of external interference.

Jamming-based AGreement (JAG). We now evaluate the performance of JAG and compare it against 2-MAG. In particular, we are interested in comparing how the percentage of positive/negative agreements and disagreement change when we increase the duration of the handshake. Intuitively, the longer t_{out} for 2-MAG and the longer t_{jam} for



(a) Local university testbed (b) Testbed in residential environment

Figure 4.17: JAG vs. 2-MAG: disagreements as function of energy with different interference patterns (a), and disagreements over time in a residential environment (b) [46].



(a) Bluetooth (b) Emulated Wi-Fi (c) Real Wi-Fi

Figure 4.18: Comparison of the rate of positive agreements and disagreements obtained by running JAG on real wireless sensor nodes, and by deriving the probabilities using the analytical model shown in Section 4.4.4 [46].

JAG, the better the performance. However, it is important to see their distribution to study the protocols' energy-efficiency and their DPA ratio under interference.

Figure 4.16 shows the results: JAG sustains a significantly lower number of disagreements compared to 2-MAG already for small values of t_{jam}. For example, 2-MAG requires more than 7.5 ms to obtain less than 1% disagreement under Bluetooth interference, whereas JAG achieves this amount with a $t_{jam} \geq 250\,\mu$s. Even though 2-MAG has a high number of positive agreements, it requires significantly higher values of t_{out} to reduce the number of disagreements and the DPA ratio. JAG, instead, has a very low rate of disagreements under every type of interference even with small t_{jam}, which enables significant energy savings, as shown in Figure 4.17(a). Furthermore, when t_{jam} is longer than the longest interference burst, we do not have any disagreements as discussed in Section 4.4.3. Obtaining this behaviour using packet-based approaches would require a significantly higher cost: Figure 4.15(e) shows that even when sending bursts of ACKs for 100 ms, one cannot still guarantee the absence of disagreements.

Finally, we validate the goodness of JAG by running a long-term experiment in our second testbed deployed in a residential environment. In particular, we compare the performance of JAG and 2-MAG over time when using $t_{jam} = 500$ μs for JAG and $t_{out} = 5$ ms for 2-MAG (Figure 4.17(b)). We do not change the configuration of the two protocols throughout the duration of the experiment. The interference in the environment changes significantly over the day: a lot of Wi-Fi activity was present during daytime in the weekend (May, 12-13), but it was quiet during night and on Monday (May, 14) during the day, as most people were not in their homes. Despite selecting a t_{out} 10 times higher than t_{jam}, JAG sustains a significantly lower number of disagreements and outperforms 2-MAG during the whole duration of the experiment.

Predictability of JAG. We now evaluate the goodness of the probabilistic model presented in Section 4.4.4 with respect to the predictability of the performance of JAG. In order to do this, we firstly obtain the *pdf* of idle and busy periods using sensor nodes in wireless sniffer mode in the different scenarios. Then, based on Equations 4.8 and 4.9, we obtain an upper bound for the probability of obtaining disagreement and a lower bound for the probability of obtaining positive agreements as a function of t_{jam} using $t_{pkt} = 1$ ms, $t_{ack} = 750$ μs, and R_{thr} = -90 dBm.

By running JAG on real wireless sensor nodes, we verify experimentally whether the probabilistic model is able to predict the performance of JAG. The results illustrated in Figure 4.18 show that our probabilistic model parametrizes correctly t_{jam} by giving an upper bound on the number of disagreements and a lower bound on the number of positive agreements, hence predicting the performance of the protocol correctly.

Chapter 5

Conclusions and Future Work

In this final chapter, we summarize the contributions of this thesis, discuss the limitations of the proposed approaches, and sketch potential for future work and further research in the covered topics.

5.1 Contributions

This doctoral thesis is devoted to the design of solutions that increase the dependability of wireless sensor networks deployed in harsh environments. In particular, we focus on indoor environments rich of radio interference and outdoor networks experiencing high temperature fluctuations, and aim to (i) characterize the environmental impact on WSN performance, and (ii) increase the reliability and availability of communication protocols in the presence of environmental changes.

Towards this goal, we first design low-cost testbed infrastructures enabling the repeatable playback of temperature variations (TempLab) and radio interference (JamLab) commonly found in real-world deployments. These low-cost extensions allow to (re)run experiments under almost identical environmental conditions and therefore play a crucial role in characterizing the environmental impact on WSN performance, and in understanding the limitations and comparing the performance of communication protocols.

We exploit TempLab – the temperature-controlled testbed that can accurately reproduce fluctuations recorded in outdoor environments – to observe and quantify the impact of on-board temperature variations on wireless sensor networks. We show that off-the-shelf low-power wireless transceivers experience a significant decrease in received signal strength at high temperatures, and capture this attenuation in signal strength as a function of temperature in a platform-independent analytical model. State-of-the-art communication protocols often neglect the attenuation that temperature has on the received signal strength in low-power wireless radios, and we use TempLab to further show that this may lead to a drastic decrease in network performance. Our experimental results indicate that (i) routing protocols can experience drastic changes in the topology of the network, including some temporary network partitions and large increases in network diameter [45], and that (ii) data link layer protocols may experience a reduced effectiveness of clear channel assessment at high temperatures that compromises the ability of a node to avoid collisions and to successfully wake-up from low-power mode. To mitigate the impact that temperature variations have on carrier sense multiple access protocols, we have developed two mechanisms that dynamically adapt the clear channel assessment threshold to temperature changes, thus making data link layer protocols temperature-aware. An extensive experimental evaluation carried out using TempLab showed that both approaches considerably increase the performance of a network in the presence of temperature variations commonly found in outdoor deployments, with up to 71% lower energy consumption and 194% higher packet reception rate.

We exploit JamLab – the low-cost infrastructure that augments existing WSN testbeds with accurate interference generation by using off-the-shelf sensor nodes – to study the performance of several state-of-the-art MAC protocols under interference. We first identify mechanisms that can improve the reliability of existing MAC protocols under interference (e.g., hand-shaking schemes preceding the actual data transmission and congestion back-off timers), and we then embed them within an existing X-MAC implementation. Our experimental results show considerable performance improvements in the presence of interference, with high packet delivery rates despite low power consumption. We further exploit JamLab to study the agreement problem in congested environments, i.e., how to agree on fundamental pieces of information such as the handover of a leader role from one node to another or the reliable exchange of channel hopping and TDMA schedules in environments with high packet loss rate. Our experimental analysis shows that traditional packet-based handshakes lead to a large fraction of disagreements in congested environments, and that excessively large energy expenditures and packet retransmissions are needed to provide performance guarantees. We tackle the problem by developing JAG [46], a protocol that uses a jamming sequence of configurable size as a last iteration of an handshake to make sure that two neighbouring nodes agree on a given piece of information. A thorough experimental analysis showed that JAG not only outperforms message-based approaches in terms of agreement probability, energy consumption, and time-to-completion, but that it can also be used to obtain performance guarantees and meet the requirements of applications with real-time constraints.

5.2 Limitations and Future Work

There are a number of limitations and potential improvements with respect to the work carried out in this thesis. We illustrate them in this section, along with a discussion on future work in a broader context.

TempLab. A first limitation of the current implementation is that the association of temperature profiles to sensor nodes is currently carried out manually by the user depending on the number of available traces and the available number of nodes in the testbed that have heating or cooling capabilities. In principle, the association of multiple traces should ideally follow the information about the original topology of the network that collected the traces (if available), in order to preserve spatial correlations across nodes. In the future, we plan to automate trace interpolation and integrate it into the testbed software.

In TempLab's current implementation, the cooling capabilities of PE nodes are sufficient to achieve a high replay accuracy, but could be further improved. To minimize the complexity of the controller, PE nodes vary the intensity of the heat lamps while the Peltier module is constantly active. One can enrich the controller with the possibility of activating or deactivating the Peltier module using the on/off wireless switch independently. This can significantly speed-up cooling and heating within the PE enclosure.

Furthermore, as we will continue using the TempLab facility in the near future, we plan to investigate if a continuous heating and cooling of the sensor nodes accelerates the ageing of the hardware (although the controllers are calibrated in order not to exceed the operating range of the individual components of each sensor platform).

Temperature-aware MAC protocols. In this thesis, we have proposed two mechanisms to dynamically adapt the clear channel assessment threshold to temperature changes, thus making data link layer protocols temperature-aware. One aspect that was not tackled in this work is how to accurately select the initial CCA threshold for the nodes in the network. The latter could indeed be chosen in such a way that the highest temperature variation expected in the environment cannot trigger a signal strength attenuation sufficient to push one of the links into the disconnected region. T'_{CCA} could also be selected to maximize energy efficiency, and in this case also the information about the interference patterns in the surroundings would be required. Knowledge about the strength and the duration of interference patterns in the surroundings would allow a node to compute how often this interference would lead to a false wake-up [187], allowing a selection that minimizes the impact of temperature variations, while maximizing energy-efficiency in the presence of external radio interference.

Another aspect not covered in this thesis is whether the solution to the signal strength attenuation problem could be also addressed at the routing layer. An alternative could be indeed to use the platform model developed in Section 3.2.3 to let the routing protocol make predictions about which paths might degrade because of temperature fluctuations without relying on current traffic statistics [117, 212].

JamLab. The current implementation of JamLab requires a manual process to iteratively select a suitable set of nodes as jammers. However, when augmenting large WSN testbeds such as Indriya [67] and TWIST [93], this process could be significantly time consuming, as it would require several experiments, as well as knowledge about the position of the nodes. We are currently working towards an automatic jammer selection method for JamLab based on simulated annealing meta-heuristic optimization, which can provide an optimal testbed configuration without the need for user interaction, and by limiting the effort to a one-time data collection [151]. We plan to add this enhancement to the JamLab tool in the future.

Comparison of different interference mitigation techniques. The design of JamLab gives the possibility to easily create benchmarks to compare the performance of different protocols under the same interference patterns. As a starting point, we have analysed the performance of several unicast MAC protocols and observed how their design can be improved. However, JamLab paves the way for an accurate comparison of protocols at different layers, as well as for a comparison of the interference mitigation techniques illustrated in Section 2.3.2, which were never properly compared against each other in the presence of different interference patterns. For example, one could study if in the presence of Wi-Fi interference it is more advisable to use backward or forward-error correction techniques, and derive a guide for system designers and developers containing hints about which of the available protocols are more suitable in a given environment.

Agreement despite interference. In this thesis, we have addressed the agreement problem by focusing on the unicast case. JAG is an agreement protocol that can give guarantees on the probability of agreement between two nodes, but the idea cannot be generalized to a multicast or broadcast scenario. For example, a parent node wishing to exchange information with all its children, would need to carry out this operation for each child individually.

Another limitation of JAG is that jamming sequences do not provide identity information, and hence may be generated by a malicious device. JAG partially solves the problem by using a mechanism to verify that the strength of the jamming signal equals the one that would be produced by the device of interest. However, security is an important concern nowadays, and it would be important to unequivocally guarantee the identity of the jamming node by means of authentication.

Impact of other environmental parameters. This thesis only focuses on temperature and radio interference, as they are the environmental parameters causing the most profound impact on the dependability of indoor- and outdoor-deployed wireless sensor networks. However, real-world deployments have shown that there are several other environmental parameters that could have a detrimental effect on the dependability of wireless sensor networks, such as humidity, foliage distribution, and rainfall, and these should also be carefully studied. One challenge is that the impact of vegetation and meteorological conditions is highly specific to the setup and location of the deployment and may require individual studies, whereas temperature variations in outdoor installations have a strong impact on the operations of all electrical and electronic components.

This thesis also did not consider changes in the radio environment that could exacerbate fading and shadowing effects. For example, the presence of people moving in proximity of the sensor nodes, could drastically change the connectivity pattern of a link, leading to a drastic decrease in performance. Another aspect not addressed in this thesis that would significantly challenge the dependability of wireless sensor networks is node mobility. In case one or more nodes in the network are mobile, achieving reliable communications in the presence of frequent environmental changes and quick variations in the connectivity to the other nodes in the network is still an open research question.

Concluding remarks. Despite the advancement that our work represents with respect to the state of the art, we are aware that the contributions described in this dissertation do not represent a definitive answer to all the problems highlighted in Chapter 2, and that achieving dependable communications in challenging environments remains a grand challenge.

Research efforts have already started looking beyond the increasingly crowded low-power radio technologies in use nowadays. In the next years, wireless sensor networks will be integrated with emerging optical and software-defined radio technologies, and will exploit new models for spectrum access that can increase the efficiency of spectrum reuse. Directional communications offered by antenna beam steering or light wave communication promise to allow a more fine-grained interference control, to increase the security from deliberate attacks, as well as to better handle the mobility of nodes, but will they be sufficient to substantially increase the dependability of low-power communications in harsh environments?

IoT applications will increasingly generate small bursts of data with stringent requirements, and the growing dependence of cyber-physical systems on communications for their control will significantly increase the need for networking protocols that can provide real-time guarantees in the years to come. Will dynamic spectrum allocation techniques, enabling wireless sensor nodes to dynamically sense the communication environment and adapt their transmission schemes in terms of waveform, spectrum access method, as well as networking protocols, actually help WSN systems to meet their quality-of-service requirements? Can recently-proposed information-centric network architectures [18, 106] be designed in such a way that they can provide a higher reliability and availability guarantees compared to host-centric architectures?

These questions are currently awaiting an answer we shall all strive to provide.

Chapter 6

Publications

This Thesis is based on the following peer-reviewed journal articles, and conference papers (ordered by publication date):

A. C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M.A. Zúñiga. Making Sensornet MAC Protocols Robust Against Interference. In Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN). Coimbra, Portugal. February 2010.

B. C.A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. The Impact of Temperature on Outdoor Industrial Sensornet Applications. In IEEE Transactions on Industrial Informatics (TII), Volume 6, Number 3, pag. 451-459. August 2010.

C. C.A. Boano, T. Voigt, C. Noda, K. Römer, and M.A. Zúñiga. JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Chicago, IL, USA. April 2011. **Best Paper Nominee.**

D. C.A. Boano, M.A. Zúñiga, K. Römer, and T. Voigt. JAG: Reliable and Predictable Wireless Agreement under External Radio Interference. In Proceedings of the 33rd IEEE International Real-Time Systems Symposium (RTSS). San Juan, Puerto Rico. December 2012.

E. C.A. Boano, H. Wennerström, M.A. Zúñiga, J. Brown, C. Keppitiyagama, F.J. Oppermann, U. Roedig, L.-Å. Nordén, T. Voigt, and K. Römer. Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers. In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom). Thórsmörk, Iceland. August 2013. **Best Paper Award.**

F. C.A. Boano, M.A. Zúñiga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Römer. TempLab: A Testbed Infrastructure to Study the Impact of Temperature on Wireless Sensor Networks. In Proceedings of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Berlin, Germany. April 2014.

G. C.A. Boano, K. Römer, and N. Tsiftes. Mitigating the Adverse Effects of Temperature on Low-Power Wireless Protocols. In Proceedings of the 11^{th} IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS). Philadelphia, PE, USA. October 2014.

Related book chapters, journal articles, and peer-reviewed papers, poster, and demos presented at International Conferences not included in this Thesis:

1. C.A. Boano, Z. He, Y. Li, T. Voigt, M.A. Zúñiga, and A. Willig. Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks. In Proceedings of the 4th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp) in conjunction with the 34th Conference on Local Computer Networks (LCN). Zurich, Switzerland. October 2009.

2. C.A. Boano, K. Römer, Z. He, T. Voigt, M.A. Zúñiga, and A. Willig. Generation of Controllable Radio Interference for Protocol Testing in Wireless Sensor Networks. In Proceedings of the 7th ACM International Conference on Embedded Networked Sensor Systems (SenSys), demo session. Berkeley, California, USA. November 2009.

3. C.A. Boano, M.A. Zúñiga, T. Voigt, A. Willig, and K. Römer. The Triangle Metric: Fast Link Quality Estimation for Mobile Wireless Sensor Networks. In Proceedings of the 19th IEEE International Conference on Computer Communications and Networks (ICCCN). Zurich, Switzerland. August 2010.

4. C.A. Boano, K. Römer, F. Österlind, and T. Voigt. Realistic Simulation of Radio Interference in COOJA. In Adjunct Proceedings of the 8th European Conference on Wireless Sensor Networks (EWSN), demo session. Bonn, Germany. February 2011.

5. C. Noda, S. Prabh, C.A. Boano, T. Voigt, and M. Alves. A Channel Quality Metric for Interference Aware Wireless Sensor Networks. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), poster session. Chicago, IL, USA. April 2011.

6. M.A. Zúñiga, I. Irzynska, J. Hauer, T. Voigt, C.A. Boano, and K. Römer. Link Quality Ranking: Getting the Best out of Unreliable Links. In Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS). Barcelona, Spain. June 2011.

7. C. Noda, S. Prabh, M. Alves, C.A. Boano, and T. Voigt. Quantifying the Channel Quality for Interference-Aware Wireless Sensor Networks. In ACM SIGBED Review - Special Issue on the 10th International Workshop on Real-Time Networks (RTN 2011), Volume 8, Issue 4, pag. 43-48. December 2011.

8. C.A. Boano, K. Römer, T. Voigt, and M.A. Zúñiga. Agreement for Wireless Sensor Networks under External Interference. In Adjunct Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN), poster session. Trento, Italy. February 2012.

9. N. Baccour, A. Koubâa, L. Mottola, M.A. Zúñiga, H. Youssef, C.A. Boano, and M. Alves. Radio Link Quality Estimation in Wireless Sensor Networks: a Survey. ACM Transactions on Sensor Networks (TOSN), Volume 8, Issue 4. November 2012.

10. N. Baccour, A. Koubâa, C.A. Boano, L. Mottola, H. Fotouhi, M. Alves, H. Youssef, M.A. Zúñiga, D. Puccinelli, T. Voigt, K. Römer, C. Noda. Radio Link Quality Estimation in Low-Power Wireless Networks. In "SpringerBriefs in Electrical and Computer Engineering – Cooperating Objects". ISBN 978-3-319-00773-1. July 2013.

11. C. Keppitiyagama, N. Tsiftes, C.A. Boano and T. Voigt. Temperature Hints for Sensornet Routing. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys), poster session. Rome, Italy. November 2013.

12. J. Brown, U. Roedig, C.A. Boano, K. Römer, and N. Tsiftes. Demo Abstract: How Temperature Affects IoT Communication. In Adjunct Proceedings of the 11th European Conference on Wireless Sensor Networks (EWSN), demo session. Oxford, United Kingdom. February 2014.

13. C.A. Boano, K. Römer, J. Brown, U. Roedig, and M.A. Zúñiga. Demo Abstract: A Testbed Infrastructure to Study the Impact of Temperature on WSN. In Proceedings of the 11th IEEE Conference on Pervasive Computing and Communications (PerCom), demo session. Budapest, Hungary. March 2014.

14. F.J. Oppermann, C.A. Boano, K. Römer. A Decade of Wireless Sensing Applications: Survey and Taxonomy. In "The Art of Wireless Sensor Networks – Volume 1", edited by H.M. Ammari. ISBN 978-3-642-40008-7, Springer. 2014.

15. J. Brown, U. Roedig, C.A. Boano, and K. Römer. Estimating Packet Reception Rate in Noisy Environments. In Proceedings of the 9th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp) in conjunction with the 39th IEEE Conference on Local Computer Networks (LCN). Edmonton, Canada. September 2014.

16. F.J. Oppermann, C.A. Boano, M. Zimmerling, and K. Römer. Automatic Configuration of Controlled Interference Experiments in Sensornet Testbeds. In Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems (SenSys), poster session. Memphis, TN, USA. November 2014.

Related technical reports and project deliverables not included in this Thesis:

1. M. Bor, K. Langendoen, C.A. Boano, F.J. Oppermann, K. Römer, A. Veiga Rico, P. Moreno Montero, R. Socorro Hernández, I. Vilajosana, M. Dohler, M. Montón, U. Roedig, A. Mauthe, G. Coulson, T. Voigt, L. Mottola, Z. He, N. Tsiftes. D-4.1 – Report on Use Case Definition and Requirements. RELYonIT project deliverable. March 2013.

2. M.A. Zúñiga, C.A. Boano, J. Brown, C. Keppitiyagama, F.J. Oppermann, P. Alcock, N. Tsiftes, U. Roedig, K. Römer, T. Voigt, and K. Langendoen. D-1.1 – Report on Environmental and Platform Models. RELYonIT project deliverable. June 2013.

3. C.A. Boano, F.J. Oppermann, K. Römer, J. Brown, U. Roedig, C. Keppitiyagama, and T. Voigt. D-4.2 – Prototype of Testbeds with Realistic Environment Effects. RELYonIT project deliverable. October 2013.

4. M. Baunach, C.A. Boano, K. Langendoen, P. Moreno Montero, M. Montón, F.J. Oppermann, U. Roedig, K. Römer, R. Socorro Hernández, T. Voigt, and M.A. Zúñiga. D-5.1 – Report on 1st Year Cooperation, Dissemination and Joint Activities. RELYonIT project deliverable. November 2013.

5. J. Brown, I.E. Bagci, U. Roedig, M.A. Zúñiga, C.A. Boano, N. Tsiftes, K. Römer, T. Voigt, and K. Langendoen. D-1.2 – Report on Learning Models Parameters. RELYonIT project deliverable. November 2013.

6. M.A. Zúñiga, F. Aslam, I. Protonotoarios, K. Langendoen, C.A. Boano, K. Römer, J. Brown, U. Roedig, N. Tsiftes, and T. Voigt. D-2.1 – Report on Optimized and Newly Designed Protocols. RELYonIT project deliverable. May 2014.

7. N. Tsiftes, T. Voigt, F. Aslam, I. Protonotarios, M.A. Zúñiga, K. Langendoen, C.A. Boano, F.J. Oppermann, K. Römer, M. Baunach, J. Brown, U. Roedig, P. Moreno Montero, R. Socorro Hernández, M. Montón, J.C. Pacho. D-4.3 – First Integrated Prototype and Experiment. RELYonIT project deliverable. May 2014.

Paper A

C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M.A. Zúñiga. **Making Sensornet MAC Protocols Robust Against Interference.** *In Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN)*, pages 272–288. Coimbra, Portugal. February 2010.

Summary. This paper describes the impact of radio interference on state-of-the-art sensornet MAC protocols. We carry out an experimental comparison of the performance of different protocols and illustrate that specific features, e.g., hand-shaking schemes preceding the actual data transmission and congestion back-off timers, play a critical role in the presence of interference. Building on top of our experimental results, we identify mechanisms that can improve the robustness of existing MAC protocols under interference, and embed them within an existing X-MAC implementation. An experimental evaluation shows that the enhanced version of X-MAC considerably improves the packet delivery rate while minimizing power consumption.

My contributions. I am one of the main authors of this paper, and wrote several sections in collaboration and discussion with the co-authors. In particular, I contributed the software used to generate interference using sensor nodes, and I carried out all the experiments in Section 5. The experiments in Section 4 were carried out together with Thiemo Voigt, and Luca Mottola. Nicolas Tsiftes played a crucial role in the implementation of the enhanced version of X-MAC. I presented the paper at EWSN'10.

©2010 Springer-Verlag Berlin Heidelberg.
ISBN: 978-3-642-11916-3.
DOI: 10.1007/978-3-642-11917-0_18.

Link:
http://link.springer.com/chapter/10.1007/978-3-642-11917-0_18.

- “Permission free of charge on this occasion does not prejudice any rights we might have to charge for reproduction of our copyrighted material in the future.”
- “You may not alter or modify the material in any manner. Abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of the author(s) and/or Springer Science + Business Media.”

Making Sensornet MAC Protocols Robust Against Interference

Carlo Alberto Boano[1], Thiemo Voigt[2], Nicolas Tsiftes[2], Luca Mottola[2], Kay Römer[1], and Marco Antonio Zúñiga[3]

[1] Institut für Technische Informatik, Universität zu Lübeck, Lübeck, Germany
[2] Swedish Institute of Computer Science (SICS), Kista, Sweden
[3] Digital Enterprise Research Institute (DERI), Galway, Ireland

Abstract. Radio interference may lead to packet losses, thus negatively affecting the performance of sensornet applications. In this paper, we experimentally assess the impact of external interference on state-of-the-art sensornet MAC protocols. Our experiments illustrate that specific features of existing protocols, e.g., hand-shaking schemes preceding the actual data transmission, play a critical role in this setting. We leverage these results by identifying mechanisms to improve the robustness of existing MAC protocols under interference. These mechanisms include the use of multiple hand-shaking attempts coupled with packet trains and suitable congestion backoff schemes to better tolerate interference. We embed these mechanisms within an existing X-MAC implementation and show that they considerably improve the packet delivery rate while keeping the power consumption at a moderate level.

1 Introduction

The increasing number of wireless devices sharing the same unlicensed ISM bands affects both reliability and robustness of sensornet communications. Sensor networks that operate, for example, in the 2.4 GHz band must compete with the communications of WLAN, Bluetooth, WirelessUSB, and other 802.15.4 devices. They may also suffer the interference caused by appliances such as microwave ovens, video-capture devices, car alarms, or baby monitors. Such problems will increase when more of these devices will be deployed in the near future.

Interference may have a deteriorating effect on communication, as it leads to packet loss and lack of connectivity. This may result in worse performance and reduced energy efficiency of sensornets, causing major issues in a number of application domains, e.g. safety-critical applications in industry and health care.

Studying the impact of interference has been hard because of the lack of proper tools that enable an inexpensive generation of controlled interference. Recently, we demonstrated a method to generate customized and repeatable interference patterns using a common CC2420 radio transceiver in special mode [1]. Using that method, we experimentally study the impact of interference on several MAC protocols, such as Contiki's NULLMAC, X-MAC, LPP, and CoReDac; and TinyOS's LPL. Our goal is to find effective mechanisms that handle interference properly. We carry out our experiments in the 2.4 GHz ISM band, which is also the most crowded one.

In this paper, we investigate which mechanisms improve the robustness of communication in congested networks while remaining reasonably energy efficient. In our experiments we identify three methods that can increase the robustness of sensornet MAC protocols against interference. Since low-power MAC protocols allow nodes to turn off their radio most of the time, they require some kind of *handshaking*. For example, in X-MAC a receiver needs to hear a strobe and answer with a strobe acknowledgment [2]. In Low Power Probing (LPP), the opposite happens: a sender waits for a probe from the intended receiver before it can send the packet [3]. Our experiments show that protocols or parameter settings that enable potentially more handshakes in case some fail due to interference are more robust. Another method that we identify is to use *packet trains* that enable the sender to quickly send multiple packets that have been accumulated during an interference period. The third method is the selection of suitable *congestion backoff schemes* when using Clear Channel Assessment (CCA) and detecting a busy channel. Based on these findings, we include these mechanisms in an X-MAC version, and show its improved robustness to interference.

Our contributions are the following. First, to the best of our knowledge, we are the first to experimentally study how interference affects different MAC protocols. Second, we identify mechanisms that enable MAC protocols to sustain high packet delivery rates while using low-power consumption even in presence of interference. Third, we show experimentally that the choice of congestion backoff schemes is critical for communication performance and energy efficiency in congested networks. Fourth, we augment an existing X-MAC implementation with these mechanisms, and demonstrate substantial performance improvements.

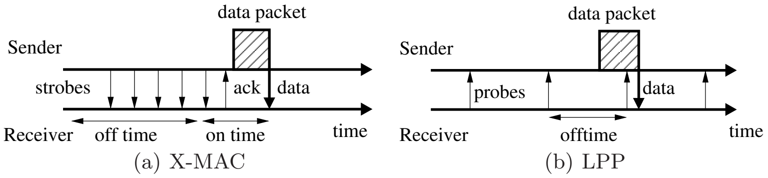
Our paper proceeds as follows. Section 2 provides an overview on the investigated MAC protocols. We describe the methodology and the setup of our experiments in Section 3. Thereafter, in Section 4 and 5, we present our experimental results and identify methods that handle interference properly. In Section 6 we design a new version of X-MAC that implements several of the identified methods and evaluate its performance. We review related work in Section 7 and present our conclusions in Section 8.

2 Background

Medium access control for wireless sensor networks has been a very active research area for the past couple of years, and the literature provides an amazing number of different implementations and incremental improvements. In our work, we exploit the four MAC layers available in Contiki (NULLMAC, X-MAC, LPP, CoReDac) and Tiny OS' LPL. Section 2.1 briefly describes these protocols, and Section 2.2 explains the role of CCA in sensornet MAC protocols.

2.1 Overview of used MAC protocols

NULLMAC. NULLMAC is a minimalistic MAC protocol that simply forwards traffic between the network layer and the radio driver. As such, it does not provide any power-saving mechanism, and keeps the radio always on. This allows



(a) X-MAC (b) LPP

Fig. 1. In X-MAC (left), the sender strobes until the receiver is awake and can receive a packet. In LPP (right), the receivers send probes to announce they are awake and ready to receive packets.

for the maximum throughput achievable, while consuming the highest amount of energy. When used with CCA and back-off timers, NULLMAC behaves as a traditional CSMA-CA protocol. Because of these characteristics, we use NULLMAC as a baseline to compare the performance of other protocols, and to verify the correctness of our setup.

X-MAC. X-MAC is a power-saving MAC protocol [2] in which senders use a sequence of short preambles (strobes) to wake up receivers. Nodes turn off the radio for most of the time to reduce idle listening. They wake up shortly at regular intervals to listen for strobes. When a receiving node wakes up and receives a strobe destined to it, it replies with an acknowledgment indicating that it is awake. After receiving the ACK, the sender transmits the data packet, as shown in in Figure 1(a).

The X-MAC implementation in Contiki has several parameters of significance to our experiments. *Ontime* determines the maximum time that a receiver listens for strobes, whereas *offtime* specifies the time to sleep between waking up to listen for strobes. *Strobe_time* denotes the duration a sender transmits strobes until it receives a strobe acknowledgment from the receiver. In the default Contiki X-MAC implementation, strobe_time = offtime + (20 × ontime).

Low-Power Probing (LPP). LPP is a power-saving MAC protocol where receivers periodically send small packets, so called probes, to announce that they are awake and ready to receive a data packet [3]. After sending a probe, the receiver keeps its radio on for a short time to listen for data packets. A node willing to send a packet turns on its radio waiting for a probe from a neighbor it wants to send to. On the reception of a probe from a potential receiver, the node sends an acknowledgment before the data packet, as shown in Figure 1(b).

The LPP implementation in Contiki contains two important parameters. *Ontime* determines how long a receiver keeps the radio on after the transmission of a probe, *offtime* is the time between probes. We use $\frac{1}{2}$ and $\frac{1}{64}$ seconds for *offtime* and *ontime* respectively. Another parameter is the time to keep an unsent packet: Contiki LPP's default value is 4×(ontime + offtime). If LPP receives a packet from the network layer when the packet queue is full, LPP discards the new packet. The queue length is configurable, and the default size is 8 packets.

Low-Power Listening (LPL). We consider a Low-Power Listening (LPL) layer that implements an asynchronous wake-up scheme for CC2420 radios [4]. Nodes periodically wake up to detect transmissions. To do so, they rely on CCA rather than attempting to pick up a full packet. Unlike X-MAC, senders repeatedly transmit the entire packet for twice the duration of the wake-up period. In case of unicast transmissions, the intended receiver may acknowledge the transmission to notify the sender on correct packet delivery so that the sender can stop transmitting earlier. To implement this functionality, packet transmissions are interleaved with periods of silence in order to allow ACK transmissions. The only LPL parameter tunable by the users is the wake-up period.

CoReDac. CoReDac is a TDMA-based convergecast protocol [5] that builds a collection tree that guarantees collision-free radio traffic. From D-MAC [6] CoReDac borrows the idea of staggered communication. To avoid collisions among packets from their children, CoReDac parents split their reception slots into subslots, and assign one to each child. Packet acknowledgments are pivotal in CoReDac because they piggyback the assignment information, and they are used for synchronizing the TDMA-schedules. A node that misses an acknowledgment must keep its radio on until it hears a new one.

2.2 Clear Channel Assessment

Clear Channel Assessment (CCA) is a mechanism used to determine if a wireless channel is currently free. In wireless MAC protocols, CCA is used to implement Carrier Sense Multiple Access: each node first listens to the medium to detect ongoing transmissions, and transmits the packet(s) only if the channel is free, thus reducing the chance of collisions. CCA is typically implemented by comparing the Received Signal Strength (RSS) obtained from the radio against a threshold. The channel is assumed to be clear if the RSS does not exceed the given threshold. As false negatives result in collisions and false positives cause increased latency, the choice of the threshold is critical [7]. When using CCA to perform CSMA, backoff schemes play an important role. There are two types of backoff: congestion backoff and contention backoff. The former controls the waiting time between consecutive assessments if the channel is not clear. The second controls the waiting time before a retransmission after a collision is detected.

3 Methodology

In our experiments, we use a set of MAC protocols from both the Contiki and TinyOS operating systems. To set a protocol's parameters, we look at the configurations used in popular, low-rate data collection applications [8, 9] that employed similar MAC protocols. These parameters are in general not set to perform optimally under interference.

3.1 Generating Controllable Interference

In our experiments we use a method proposed by Boano et al. [1] to generate customized, controllable, and repeatable interference patterns using common sensornet devices. This method enables the generation of precisely adjustable levels of interference on a specific channel, by exploiting the special test modes of the radio chip.

3.2 Performance Measurements

We use Contiki's software-based power profiler [10] to measure power consumption. For the experiments concerning TinyOS, we have implemented the same mechanism in TinyOS. For computing the power consumption, we assume a current of 20 mA for the radio in receive mode, and a voltage of 3 V, as measured by Dunkels et al. [10]. In all our experiments, the power consumed by the radio in receive mode (*RX power*) is much higher than the one used for transmitting (*TX power*). Because of its strobe mechanism, X-MAC has the highest TX power among the MAC protocols that we examine. At 60% interference, the TX power is around 1 mW, whereas the RX power is almost 20 mW. For LPP instead, the TX power is usually between 0.1 and 0.2 mW only. The power values represent the average power during the full experiment. Since the RX power is at least an order of magnitude larger than the TX power in our experiments, we display only the RX power in our graphs.

3.3 Experimental Setup and Interference Model

In our experiments we put three nodes near each other: a sender, a receiver, and an interferer. The latter interferes using the CC2420's maximum output power level (31), while the sender and the receiver use TX power level 7. The placement of the nodes and their power levels ensure that an active interferer blocks any ongoing communication between the sender and the receiver.

Interference may result from other packet radios (Wi-Fi, Bluetooth, and other sensor networks) operating in the same frequency band, and from other electromagnetic sources such as motors or microwave ovens. Unfortunately, at the time of writing, there are no accepted interference models – an important research issue by itself that is beyond the scope of this paper. Hence, we resort to two simple models here. The *bursty interferer* models continuous blocks of interference with uniformly distributed duration and spacing. This type of interference may be caused, for example, by Wi-Fi or Bluetooth transmissions. The *semi-periodic interferer* also models continuous blocks of interference, but the duration of the periods and their spacing have smaller variance. This type of interference may be caused, for example, by a sensornet performing periodic data collection.

Bursty Interference. In order to describe the transmission and interference patterns, let us define the following random variables:

- S: Bernoulli random variable with parameter 0.5;
- R: Uniformly distributed over [0, 100];
- $Q(x)$: Uniformly distributed over [0, x].

Interference follows continuous off/on periods, and is dictated by a simple two-state discrete Markov process, as depicted in Figure 2. C denotes the clear

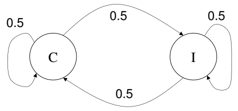


Fig. 2. The interference model used in our experiments.

channel state, and I denotes the interference state. The transitions between the two states is specified by S. At each step of the Markov process, we obtain a time period, $R \times Q(x)$, that determines the duration of the next state. For example, assuming that we move to state I and that we obtain values $R = 40$ and $Q = 20$, the next period will be an interference period of length $40 \times 20 \times 0.3$ ms=240 ms (0.3 ms is a constant factor). $Q(x)$ is used to scale the burstiness of the interference. A higher value represents longer interference slots, such as the ones caused by bursts of Bluetooth or Wi-Fi traffic, whereas a lower value represents shorter transmissions. In the experiments we will select a configuration with long interference slots ($x = 50$) that we call *long bursts*, and a configuration with shorter slots ($x = 8$) that we call *short bursts*.

Semi-Periodic Interference. The semi-periodic interferer is a 2-stage process. As described above, we have a clear channel C and an interference I states. The process stays in state I for a time that is uniformly distributed between $\frac{9}{16}$ seconds and $\frac{15}{16}$ seconds. After the transition to state C, it stays in this state for a time that is uniformly distributed between $\frac{3}{4} \times$clear_time and $\frac{5}{4} \times$clear_time, where *clear_time* is a parameter that determines the rate of interference.

4 Experimental Evaluation: the Performance of MAC Protocols under Interference

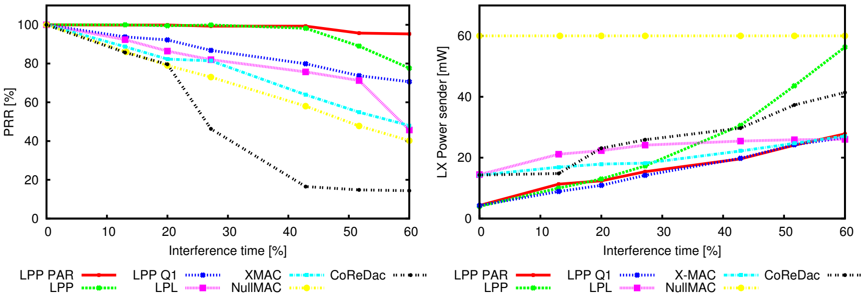
In this section we report on the performance of several MAC protocols under the different interference patterns described in the previous section.

4.1 Semi-periodic Interference

In our experiments, the sender transmits unicast packets with a payload of 22 bytes to the receiver in a time uniformly distributed between 0.75 s and 1.25 s. We collect the measurements until several thousands packets have been transmitted. We use a semi-periodic interference pattern as described in Section 3.

Figure 3 shows the results of our experiments with different MAC protocols tested against varying interference rates. As expected, the PRR in NULLMAC decreases linearly with the interference rate, following the rule 100% minus the interference rate, which is the probability that a packet is not interfered (Figure 3(a)). The RX power consumption when using NULLMAC is 60 mW independently on the interference pattern, since NULLMAC keeps the radio always on (Figure 3(b)). This confirms the validity of our setup, described in Section 3.

Figure 3(a) shows that all variants of LPP have fairly high packet reception rates compared to the other protocols we consider. Among LPP-based solutions,



(a) Packet Reception Rate (b) RX Power Consumption (sender side)

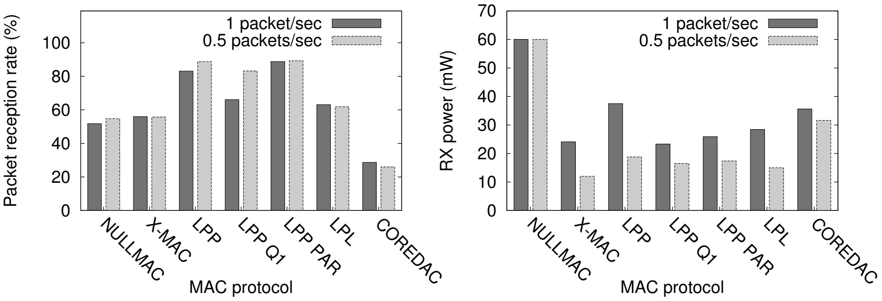
Fig. 3. MAC protocols performance under semi-periodic interference.

the best performance is obtained with LPP-PAR, where the receiver transmits a new probe immediately after a packet reception. By doing so, the sender can drain its queue when the interference clears and sustain a high PRR also under high interference by deferring transmissions until interference is over. LPP-PAR outperforms both the standard LPP version, and the so called LPP-Q1, that does not have a queue: a new packet from the upper is discarded in case the previous one has not been transmitted by the MAC layer. At an interference rate of 42%, LPP-Q1 still achieves a PRR of about 80%, showing that even only two probe attempts provide more opportunities to deliver a packet than other solutions.

Figure 3(b) shows that the power consumption of LPP-Q1 is lower than the standard LPP one. The reason comes from the lower PRR shown by LPP-Q1: with fewer packets to be transmitted, the radio is turned off more often. This difference becomes very apparent at an interference rate of 60%, where LPP has its radio turned on almost all the time since there is almost always a packet in the queue waiting to be transmitted. In contrast with the default LPP, LPP-PAR can quickly drain its queue during interference-free periods and hence turn off quickly its radio, saving a substantial amount of power.

X-MAC's packet reception rate is similar but slightly higher than NULL-MAC's (Figure 3(a)), since in X-MAC the sender's strobe_time is a little longer than the receiver's off_time. Hence, the receiver has in average more than one chance to hear a strobe. Furthermore, under a semi-periodic interference pattern, it is unlikely that interference comes into effect during the exchange of strobe, acknowledgment, and data packet, which take very little time. Therefore, if the strobe succeeds, the entire operation most likely successfully completes. The same reasoning also applies for CoReDac when the interference rate is 20% or lower. At higher interference, however, CoReDac looses synchronization and its performance drastically degrades.

With regard to LPL, we observe two modes of operations along the PRR axis in Figure 3(a). When the interference rate is lower than 60%, the CCA



(a) Packet Reception Rate

(b) RX Power Consumption (sender side)

Fig. 4. MAC protocols performance under bursty interference.

mechanism is reasonably effective at detecting the presence of interference, and packet losses occur mostly because of data corruption during the transmission. Indeed, we verify that an increasing number of packets are received but do not pass the integrity checks. The increasing power consumption shown for LPL in Figure 3(b) is simply an effect of the decreasing PRR: the fewer packets are received, the less likely is the sender to receive the acknowledgment and stop the transmissions earlier. On the other hand, at 60% interference it is often the case that the CCA mechanism never finds the channel free. After a maximum number of reattempts, the packet is dropped on the sender side, causing a drastic decrease in PRR. However, without even transmitting the packet, not much energy is spent on the sender side. This is confirmed in Figure 3(b), where the power consumption at 60% interference is still comparable to other settings.

Our results suggest that more handshakes opportunities improve the PRR in interfered networks. When comparing different LPP versions with each other, we can see that we can achieve a low power consumption and a high PRR using LPP-PAR, thanks to its queue drain when a period of interference has ended.

Impact of Queue Size on Performance. Our experiments clearly show that the queue size may drastically change the performance of a MAC protocol under interference. We investigated the impact of the queue size both on power consumption and packet reception rate by running LPP with different queue sizes under 60% semi-periodic interference. Our results show that a queue size of four packets guarantees good performance.

4.2 Bursty Interference

We carry out the same set of experiments in presence of bursty interference ($x = 50$, see Section 3), and different transmission rates, in order to investigate how performance changes depending on the network load. Figure 4 illustrates the results.

For most MAC protocols the PRR does not change depending on the transmission rate (Figure 4(a)). In most cases, indeed, the interference rate is what ultimately determines the observed PRR. An exception is LPP-Q1, where the PRR increases by almost 10% when the application transmits packets less frequently. The reason is that with higher transmission rates, a packet cannot be sent before the application hands the next packet to the MAC layer, and thus the latter packet is discarded. This can either happen with long periods of interference, or when periods of interference overlap with the instants in which the receiver sends probes.

5 The Impact of Clear Channel Assessment and Congestion Backoff under Interference

While many contention-based MAC protocols implement CSMA, one could also start transmitting a packet without carrying out CCA. The latter approach saves the CCA overhead of listening to the channel and switching the radio between send and receive modes, which may take hundreds of microseconds [11]. Few retransmissions consume a negligible amount of power compared to a continuous use of CCA. An increased probability of collisions may be negligible in low data rate applications, but not in settings with high interference.

A second aspect that affects the performance of CSMA-based MAC protocols such as B-MAC [12], WiseMAC [13], and BoX-MAC [14] is the backoff algorithm that adapts the scheduling of CCA executions to wireless channel conditions. B-MAC, for example, uses by default a small random congestion and contention backoff time, but does also support user-defined backoff schemes. BoX-MAC uses a randomized long congestion backoff period in the order of a few hundred milliseconds.

In this section we identify (1) the scenarios where adopting CCA improves or decreases the performance of MAC protocols under interference, and (2) if the choice of the congestion backoff scheme plays a pivotal role under interference. We investigate these issues in terms of energy efficiency and latency.

5.1 Experimental Setup

In our first experiment, we compare a scenario in which CCA is not used (and packets are sent without a carrier sense) with one in which a node sleeps after detecting a busy channel for a congestion backoff time B_C. We explore different types of backoff algorithms, in particular null (no waiting time), constant (waiting time uniformly drawn from a fixed backoff window), linear (backoff window increases by a constant amount after failed CCA), quadratic (backoff window squared after failed CCA), and cubic (backoff window cubed after failed CCA).

We select an initial backoff time randomly short and we eventually increase it according to the backoff algorithm. We further study a variant where the backoff is truncated after $R = 8$ CCA attempts. We use the CC2420's default CCA threshold.

Our experimental setup is described in Section 3. The transmitter sends N packets towards the receiver at different transmission rates. Each packet has to be acknowledged within $\frac{1}{64}$ seconds.

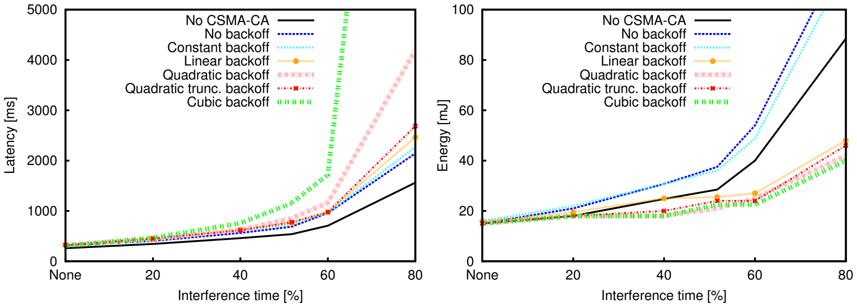
We further investigate two different strategies for scheduling retransmissions. With the first approach, queued packets are retransmitted immediately after timeout. With the second approach, the sender turns off its radio after a timeout occurs, and the queued packets are retransmitted according to the original packet transmission rate (e.g. after 0.5 seconds if we transmit 2 packets per second). We measure the latency required to transmit the sequence of N packets and the total amount of energy consumed by the radio of the sender. The latter is appropriate because interference mainly affects the sender, assuming that a receiver can distinguish valid data from interference and go back to sleep in case of the latter. The sender node runs NULLMAC with or without CSMA, and its radio is turned off after the reception of an ACK (or after the timeout fires), and turned on again for the next transmission. Since we are only interested in the energy consumption of the sender, the receiver keeps the radio on all the time. To isolate the effect of CCA from that of other MAC mechanisms, we avoid mechanisms such as LPL and the associated use of long preambles.

5.2 Experimental Results

In the first set of experiments, we evaluate the communication performance when transmitting $N = 50$ packets at the highest available rate, and compare transmissions with and without CSMA. We average the results after sending several thousand packets. Figure 5 shows the results. As expected, the more aggressive the backoff strategy is, the lower is the energy required to complete the transmission. The latency increases proportionally with the backoff delays, however, indicating a tradeoff between energy consumption and latency. The energy consumption is, however, significantly reduced when not using CSMA, but using aggressive backoffs such as quadratic and cubic algorithms on a channel that is interfered more than 20% of the time. We can also see that truncating the backoff window yields a good balance between energy and latency.

In the scenario presented above, the packets are retransmitted as soon as the timeout event occurs. If queued packets are retransmitted back-to-back under interference, there is a significant waste of energy due to the medium still being busy, while a retransmission based on the original transmission rate increases the overall latency. To quantify these issues, we carry out another experiment with different periodic transmission rates. We transmit bursts of $N = 10$ packets with and without CSMA, using null, linear, and quadratic congestion backoff schemes. Then we apply a bursty interference pattern with long bursts ($x = 50$) and measure the latency and energy consumption at the sender side, averaging the results of several hundred bursts.

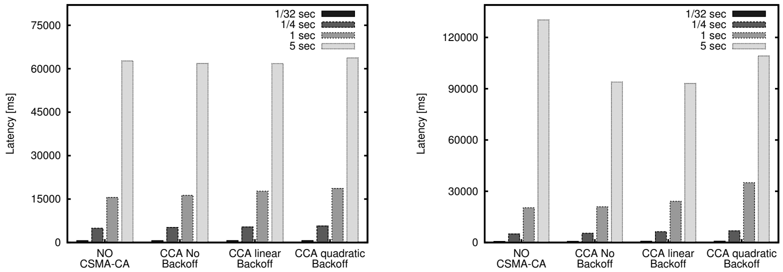
Figures 6 and 7 show the results. As expected, if queued packets are retransmitted back-to-back, the approach without CSMA performs poorly. A configuration with quadratic congestion backoff requires only 5% of the energy used without CSMA with an acceptable latency because of the fewer attempts. If,



(a) Latency

(b) Energy consumption

Fig. 5. Energy consumption and latency measured at the sender side, when sending bursts of $N = 50$ packets at the highest rate available.



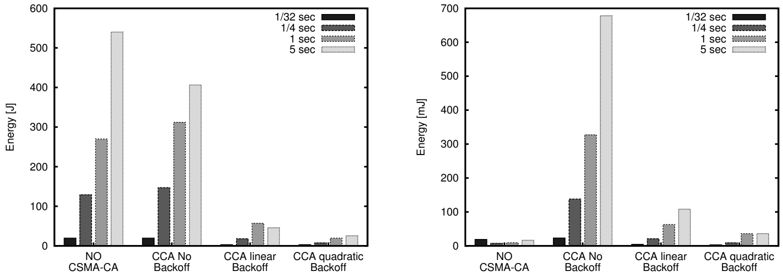
(a) Immediate retransmission

(b) Delayed retransmission

Fig. 6. Latency measured at the sender side when sending bursts of $N = 10$ packets at different transmission rates, with different retransmissions schemes.

instead, queued packets are retransmitted according to the original transmission rate, the protocol that does not adopt CSMA performs better in terms of energy efficiency. This is because it attempts to transmit only at the instants defined by the transmission rate, while the approach with CSMA and backoff tries to find the first instant at which the medium is free, often without success. This makes the approach without CSMA more energy-efficient, but comes with an increased latency when sending at low transmission rates, such as one packet every 5 seconds w.r.t. CSMA transmissions. As in the previous experiment, a more aggressive congestion backoff scheme such as the quadratic algorithm shows a good balance between latency and energy consumption.

In addition to the above experiments with long bursts, we also carried out experiments with shorter bursts ($x = 8$, see Section 3). Due to space constraints we do not show the results here. These experiments indicate a better performance of protocols using CSMA, because shorter slots will imply a lower energy consumption since the channel will be sampled a smaller amount of times.



(a) Immediate retransmission (b) Delayed retransmission

Fig. 7. Energy consumption measured at the sender side, when sending bursts of $N = 10$ packets at different transmission rates, with different retransmissions schemes.

In conclusion, our experiments demonstrate that the choice of congestion backoff scheme plays a pivotal role for MAC protocols that use CCA. These results act as a guideline for protocol designers. A CSMA approach with a quadratic backoff –truncated or not– performs well in most scenarios.

6 Improvements

The results presented in Section 4 show two methods that can make MAC protocols more robust against interference: (1) holding a packet longer so that multiple handshake attempts are possible, and (2) implementing packet trains as a means to quickly send multiple packets that have accumulated during interference. Section 5 further shows that the power consumption can be reduced by applying suitable congestion backoff schemes when using CCA. We extend the X-MAC implementation in Contiki 2.3 with these mechanisms, and evaluate it under random interference patterns.

6.1 Design and Implementation of a Robust X-MAC

We design a new version of X-MAC, called X-MAC/Q, that is able to maintain high packet reception rates and low power consumption despite being challenged by interference. The new version contains a packet queue implemented by using a statically allocated array of packets and their corresponding attributes. By default, the queue stores up to four packets, the optimal value for LPP as discussed in Section 4.1. Since only unicast packets are acknowledged in the X-MAC protocol implementation, we only queue unicast packets.

Packet Queue with Fast Drain. Unlike the original implementation of X-MAC in Contiki, our augmented implementation revolves around the packet queue. This distinction starts from the existing packet transmission method, *qsend_packet()*, where all unicast packets are put into a queue. The packets will not be sent directly, but instead linger shortly for a configurable time ($\frac{1}{32}$ s

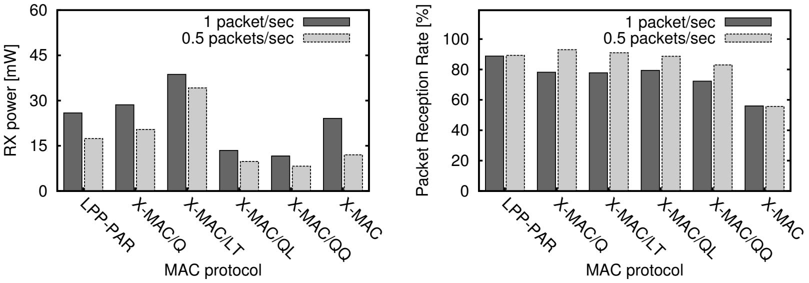


Fig. 8. Our experiments show that the proposed mechanisms increase the robustness of X-MAC to interference.

in our experiments.) The linger time makes it possible to accumulate packets into the queue, which allows the layer on top of X-MAC to create a burst of packets. When the accumulation timer has expired, X-MAC/Q gets the oldest packet from the queue, and immediately starts sending strobes to the addressed receiver of the packet. To enable fast queue draining, each strobe contains the amount of packets for the destination that the sender has in its queue. If the sender receives a strobe acknowledgment within a configured waiting time, it sends one packet at a time, including the strobe procedure, separated by a very short time ($\frac{1}{128}$ s) instead of the usual duty-cycle interval. If the sender does not receive the strobe acknowledgment, a new attempt comes after $\frac{1}{32}$ s. Packets are removed from the queue when they have either been successfully sent, or timed out after 10 s. The X-MAC reception method requires only two changes. First, each received strobe will contain the amount of packets x that the receiver should receive in a train. Second, the receiver stays awake until it has received x packets since the strobe.

Clear Channel Assessment with Congestion Backoff. Based on the results in Section 5, we extend X-MAC/Q to include clear channel assessments with a linear and a quadratic congestion backoff timers. The version with the linear backoff is called X-MAC/QL, whereas the version with quadratic backoff is called X-MAC/QQ. Before sending out the first strobe the new versions turn on the CCA to check if the channel is clear. If the CCA check fails, we wait for ($\frac{1}{128} \times$ number_of_attempts) or ($\frac{1}{128} \times$number_of_attempts^2) milliseconds before another attempt for X-MAC/QL and X-MAC/QQ respectively.

6.2 Experimental Evaluation

We repeat the experiments with the bursty interferer described in Section 4.2 using our improved versions of X-MAC. For comparison, we also show the LPP-PAR and another X-MAC improvement that we call X-MAC/LT. X-MAC/LT is similar to X-MAC except for one parameter, *strobe_time*, which we increase from offtime + 20 × ontime to 4 × offtime + 20 × ontime. Because X-MAC/LT holds packets longer, we expect a higher PRR compared to X-MAC.

Figure 8 shows that both X-MAC/Q and X-MAC/LT significantly increase the PRR compared to the default X-MAC. When the applications send one packet every two seconds, the PRR is similar to the one of LPP-PAR. Also, both new X-MAC versions show a similar rate, but the left graph in Figure 8 shows that the power consumption is much higher for X-MAC/LT than for X-MAC/Q. X-MAC/QQ and X-MAC/QL achieve a good PRR with very low power consumption. Since both protocols wait for an increasing amount of time when the medium is kept busy, they send less strobes and avoid to wait for strobe acknowledgments that will not arrive, thus saving a significant amount of power. Compared to X-MAC/QQ, X-MAC/QL consumes slightly more energy but achieves a higher PRR. This follows the results presented in Section 5.2: the linear backoff causes more frequent samples of the channel than the quadratic one does, leading to higher power consumption. On the other hand, the quadratic algorithm may grow its sampling interval exponentially up to a point where expired packets will be removed from the queue.

In all our experiments, we set the protocol parameters based on the configurations of similar MAC protocols in popular applications [8, 9], since our goal is not to optimize parameters but to identify mechanisms that enable good performance during interference. One way of increasing the handshake frequency would be to change the parameters. In X-MAC, this is the *offtime* parameter. We have rerun the same experiment as in Figure 8, but halved the *offtime* to 1/4 s for X-MAC and X-MAC/Q. Our results show similar improvements in PRR and power consumption for both protocols. For the CCA versions with a linear backoff, the improvements of the PRR were smaller but the power consumption was decreased by around 40%.

In summary, our results show significant improvements of the packet reception rate for X-MAC/Q with a moderate increase in power consumption. X-MAC/QQ and X-MAC/QL's power consumption is even lower than X-MAC's despite that they achieve a much higher PRR.

7 Related Work

Radio interference has been a topic of significant interest in the sensor network community. Most of the earlier work focused on deriving fair transmission schedules by synchronizing the transmission of neighboring nodes in the presence of interference [15–18]. Our work also addresses MAC performance, but our goal is to identify experimentally some mechanisms that improve the robustness of MAC protocols against interference.

Zhou et al. present some important differences between the interference behavior of real and ideal scenarios [19, 20]. Others study interference effects on real deployments: Rangwala et al. propose an interference-aware fair-rate control evaluated on real hardware [21]. Others have proposed frequency hopping solutions for 802.15.4 networks in order to overcome Wi-Fi interference [22, 23].

Motivated by the empirical works mentioned above, we (1) analyze experimentally the impact of interference on various MAC protocols, and (2) propose mechanisms to increase packet delivery rate and reduce energy consumption.

An important group of work pertaining to this study is the set of notable MAC protocols evaluated on empirical testbeds, in particular X-MAC[2], LPP[3], LPL[12]. Most of these evaluations focused on energy efficiency and delay under different traffic patterns while we evaluate the protocols behaviour under various degrees of interference. Bertocco et al. investigate efficient CCA thresholds in presence of in-channel wide-band additive white Gaussian noise [7]. In this work, we study the role of CCA and congestion backoff schemes with respect to energy consumption and latency under generic patterns of interference. So far, thorough studies on backoff schemes have been performed only with respect to contention resolution [24], [25], and [26], where Jamieson et al. propose a MAC protocol that uses a fixed-size contention window and a non-uniform probability distribution of transmitting in each slot within the window.

Moss and Levis envisioned how a long congestion backoff could at the same time optimize energy and delivery rates in congested networks [14]. However, they do not determine optimal backoff periods and do not quantify the effects of different schemes. We demonstrate experimentally the impact of the congestion backoff time on energy efficiency and latency in networks with high interference.

8 Conclusions

In this paper, we experimentally study the impact of interference on several MAC protocols. Using the results from our experiments, we identify mechanisms that make MAC protocols more robust against interference. We augment an existing X-MAC implementation with these mechanisms, and demonstrate improved packet reception rates and reduced power consumption in cases where the radio communication is challenged by interference.

Acknowledgments

This work has been partially supported by the European Commission under the contract No. FP7-2007-2-224053 (CONET). This research has been also partially financed by VINNOVA, the Swedish Agency for Innovation Systems, by SSF, and by the IRCSET Postdoctoral fellow Grant PD200857, SFI Grant No. SFI08CEI1380. This work has been partially supported by the Cluster of Excellence 306/1 "Inflammation at Interfaces" (Excellence Initiative, Germany).

References

1. C.A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig. Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks. In *Proc. of the 4th Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, Zurich, Switzerland, October 2009. IEEE Computer Society.
2. M. Buettner, V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *ACM SenSys*, 2006.
3. R. Musaloiu-E., C-J. M. Liang, and A. Terzis. Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. In *IPSN '08*, 2008.

4. TinyOS Community Forum. TinyOS TEP 126 - CC2420 radio stack. www.tinyos.net/tinyos-2.x/doc/html/tep126.html.
5. T. Voigt and F. Österlind. CoReDac: Collision-free command-response data collection. In *13th IEEE Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, September 2008.
6. G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *International Parallel and Distributed Processing Symposium (IPDPS)*, 2004.
7. M. Bertocco, G. Gamba, and A. Sona. Experimental Optimization of CCA Thresholds in Wireless Sensor Networks in the Presence of Interference. In *Proc. of IEEE Workshop on ElectroMagnetic Compatibility (IEEE EMC)*, June 2007.
8. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, September 2002.
9. G. Tolle et al. A macroscope in the redwoods. In *SenSys*, pages 51–63, 2005.
10. A. Dunkels, F. Österlind, 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)*, Cork, Ireland, June 2007.
11. K. K. Chintalapudi and L. Venkatraman. On the design of mac protocols for low-latency hard real-time discrete control applications over 802.15.4 hardware. In *the 7th Conf. on Information Processing in Sensor Networks (IPSN)*, April 2008.
12. J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *ACM SenSys*, 2004.
13. A. El-Hoiydi and J.D. Decotignie. Wisemac: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *ISCC*, June 2004.
14. D. Moss and P. Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. Technical Report SING-08-00, Stanford University, 2002.
15. L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(1):163–173, 2005.
16. L. Chen, SH Low, and JC Doyle. Joint congestion control and media access control design for ad hoc wireless networks. In *INFOCOM*, 2005.
17. Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *IEEE/ACM Transactions On Networking*, 15(1):133–144, 2007.
18. Y. Yi, G. de Veciana, and S. Shakkottai. On optimal MAC scheduling with physical interference. In *INFOCOM*, 2007.
19. G. Zhou, T. He, JA Stankovic, and T. Abdelzaher. RID: Radio interference detection in wireless sensor networks. In *INFOCOM*, 2005.
20. G. Zhou et al. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.
21. S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *ACM SIGCOMM*, 2006.
22. R. Musaloiu-E. and A. Terzis. Minimising the effect of wifi interference in 802.15.4 wireless sensor networks. *Int. Journal of Sensor Networks*, 3:43–54, Dec. 2007.
23. J. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *EWSN*, February 2009.
24. Z. Yuan et al. A backoff copying scheme for contention resolution in wireless sensor networks. In *Wintech*, September 2009.
25. A. Athanasopoulos et al. 802.15.4: The effect of different back-off schemes on power and qos characteristics. *IEEE ICWMC*, 2007.
26. K. Jamieson, H. Balakrishnan, and Y.C. Tay. Sift: a mac protocol for event-driven wireless sensor networks. In *EWSN*, February 2006.

Paper B

C.A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. **The Impact of Temperature on Outdoor Industrial Sensornet Applications.** *In IEEE Transactions on Industrial Informatics (TII), Volume 6, Number 3, pages 451–459.* August 2010.

Summary. This article takes as a case study the deployment of a wireless sensor network in an oil refinery in Portugal, in which sensor nodes are deployed outdoors and experience high temperature fluctuations. First, this article shows experimentally that temperature directly affects the communication between sensor nodes, with a significant decrease in the strength of the wireless signal at high temperatures. We describe how this may affect the design of applications and communication protocols that must operate outdoors, and further show the impact that specific implementation requirements, such as the the enclosure of nodes in ATEX fire-safe casing, have on low-power communication. Second, this article shows that it is possible to decrease the transmission power when operating at low temperatures without affecting performance, leading to a significant increase in network lifetime. In view of our experimental results, this article finally elaborates on how the temperature influences both the design and the deployment of wireless sensor networks in industrial environments.

My contributions. I am the main author of this article and I carried out the experiments showing the impact of temperature on WSN performance. I wrote the vast majority of the paper in collaboration and discussion with the co-authors, who significantly helped in elaborating, in view of the experimental results, how the temperature influences both the design and the deployment of wireless sensor networks in industrial environments. Nicolas Tsiftes helped in measuring the current consumption of the CC2420 radio chip experimentally (Section V-B).

©2010 IEEE.
ISSN: 1551-3203.
DOI: 10.1109/TII.2009.2035111.

Link: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5345684.

2. Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line;
3. In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink.

The Impact of Temperature on Outdoor Industrial Sensornet Applications

Carlo Alberto Boano[†¶], Nicolas Tsiftes[†], Thiemo Voigt[†], James Brown[††], and Utz Roedig[††]

***Abstract*—Wireless sensor networks are being considered for use in industrial process and control environments. Unlike traditional deployment scenarios for sensor networks, in which energy preservation is the main design principle, industrial environments stress worker safety and uninterrupted production. To fulfill these requirements, sensor networks must be able to provide performance guarantees for radio communication.**

In this article, we consider as a case study the deployment of a sensornet in an oil refinery in Portugal, where sensor nodes are deployed outdoors and might experience high temperature fluctuations. We investigate how the variations of ambient temperature influence data delivery performance and link quality in low-power radio communications. We also study the impact that specific implementation requirements, such as the ATEX fire-safety regulations, can have on the design of the overall network.

Our experiments show that temperature directly affects the communication between sensor nodes, and that significantly less transmission power is required at low temperatures. We further illustrate that it is possible to save up to 16% energy during nights and cold periods of the year, while still ensuring reliable communication among sensor nodes. In view of these experimental results, we elaborate on how the temperature influences both the design and the deployment of wireless sensor networks in industrial environments.

***Index Terms*—Electrical equipment enclosures, Energy conservation, Estimation, Industrial control, Petroleum industry, Power demand, Radio communication, Temperature, Wireless sensor networks.**

I. Introduction

Wireless sensor networks are successfully used for applications such as precision agriculture, military surveillance, and environmental monitoring. Recently, sensornets have been considered for use in industrial control and process automation applications because of the benefits obtained from wireless deployment: reduced costs and increased system flexibility.

To support this application domain, sensor networks must assure a certain data transport delay bound and a certain degree of reliability. Unfortunately, most sensor network protocols are designed to preserve energy rather than to meet performance guarantees. Hence, it is necessary to develop new protocols and mechanisms for sensornets that are able to give performance assurances while remaining reasonably energy efficient.

A sensor network used for process automation and control must be able to deal with fluctuating channels and environmental characteristics. For example, a communication protocol should be able to maintain a requested packet delivery rate also when link reliability drops for a while. When sensor nodes are deployed outdoors, the fluctuations might be high because of changes in weather conditions or in the environment. To be able to design and build protocols that can compensate and deal with varying conditions, the dynamics of channel fluctuations must be characterized.

In this article, we investigate how ambient temperature and weather conditions affect link quality and data delivery in low-power wireless communication. We focus our study on a sensor network deployment in an oil refinery in Portugal. In this context nodes have to be deployed outdoors and must be encased in ATEX-compliant boxes [1] to meet EU fire safety regulations. We investigate the impact that temperature variations and ATEX casing have on the design of sensor networks. Furthermore, we characterize how the energy consumption of sensor nodes is affected by temperature.

Our contributions are threefold. First, we provide experimental results that show how temperature fluctuations can create a significant variation of signal strength of up to 10 dBm. We describe how this may affect the design of applications and communication protocols that must operate outdoors. Second, we show the impact that the introduction of the ATEX casing has on low-power communication. Third, we show that temperature indirectly affects the overall energy consumption of sensor nodes. We also show that it is possible to decrease the transmission power when operating at low temperatures. Thus, nodes can save up to 16% of the power spent for transmitting packets and consequently system lifetime can be improved. In order to evaluate precisely the amount of energy saved, we measured the current consumption of all 32 available output power settings of the widely used CC2420 radio chip. We show that the measured current consumption differs from values commonly used in existing literature.

This article proceeds as follows: Section II provides a description of the application context of our case study. We quantify the impact of temperature on link quality in Section III. Thereafter, we analyze the impact of ATEX-compliant casing on temperature and link quality in Section IV. In Section V we show how the temperature influences the transmission power needed to maintain network connectivity. In Section VI we discuss how the dependency between temperature and link quality affects the network design in our investigated application scenario. After an overview of the related work in Section VII, we conclude the article in Section VIII.

II. Application Context

Process automation and control applications have stringent requirements on data transport delay and reliability. In order to understand how such systems operate and the requirements they must meet, we carried out a case study in the context of the GINSENG [2] project. We investigate a wireless sensor

[†]Swedish Institute of Computer Science, Kista, Sweden. [††]Lancaster University Computing Department, Lancaster, UK. [¶]Institut für Technische Informatik, Universität zu Lübeck, Lübeck, Germany. E-mail: {cboano, nvt, thiemo}@sics.se, {j.brown, u.roedig}@lancaster.ac.uk, cboano@iti.uni-luebeck.de.

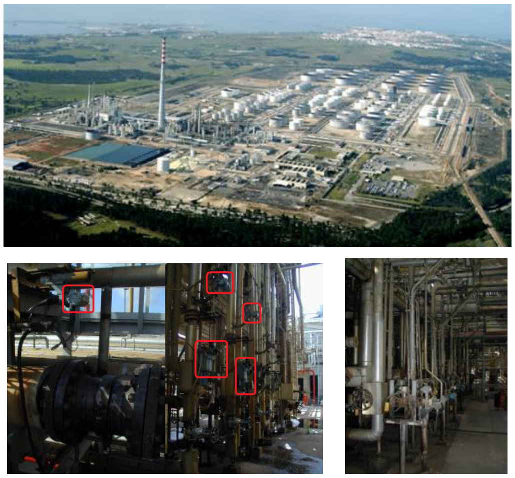


Fig. 1. The GALP oil refinery in Sines, Portugal [3] is a complex industrial facility with more than 35,000 sensors and actuators installed.

network deployed for process control and automation in the petrochemical industry.

The GALP [3] oil refinery at Sines, Portugal (see Figure 1) is a complex industrial facility that includes a wide range of processes that must be carefully monitored and controlled. Health and safety are of utmost importance in this environment: fire prevention, safe operation of machinery, and careful handling of products have to be considered when designing a sensornet for such an environment.

A. The Refinery Monitoring and Control System

There are currently 35,000 sensors and actuators in use in the refinery to perform real-time monitoring of industrial operations such as leakage detection, measurement of pressure in the pipes, and control of fluid levels. The extensive monitoring of the refinery provides essential information to ensure a good health of its production processes. In the oil refinery there are 3 systems for monitoring and control the plant: the *indicatory system*, the *control system*, and the *emergency system*.

The *indicatory system* is used to provide the control center with information about status and faults of the equipment, as well as general aspects of the environment. Within this system, information flows one way from the in-field sensors to the control center. Here, the sensor data is typically not vital, but should reach the control center to inform the operators of potential dangers.

The *control system* is used to control different aspects of the refinery. Information flows in both directions: from in-field sensors to the control center, and from the control center to actuators. In this system it is important that data arrives at its intended destination quickly and reliably. Operators require instant feedback from the sensors because the actuators are used to control equipment.

The *emergency system* is used to monitor and control mission critical systems, and to trigger alarms in order to prevent an accident. Sensors and actuators in this system are part of a closed loop system without user intervention. The information flowing in these systems is vital, and thus requires the highest level of reliability and the lowest delay bounds.

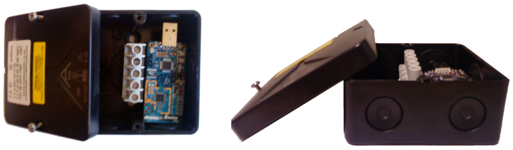


Fig. 2. A Sentilla Tmote Sky node inside an ATEX-compliant enclosure.

B. Challenges of a Wireless Monitoring and Control System

Most sensors and actuators in the oil refinery use wired technologies such as 4-20 mA systems. In such industrial environments, the work required to deploy new sensors can be very expensive. Because of their flexibility, wireless sensor networks can be employed to ease and reduce the cost of deployment. At the same time, they must assure the same performance as their wired counterparts do.

When deploying sensors in any industrial setting it is important to consider the environment in which they will be deployed. In the context of the refinery, the sensors will be deployed mostly outdoors and they must meet a number of industrial regulations. Because nodes are deployed outdoors they are exposed to changing weather conditions and, consequently, changing link quality. Temperature changes may affect the link quality as well, so it is important to quantify these effects before designing a sensor network.

The oil refinery deployment further restricts the network design because of its potentially explosive atmosphere. The European Union regulates the equipment used in such contexts as specified in the ATEX directives [1], ensuring that the equipment is not a potential source of ignition. Although it is possible to obtain ATEX certification for a sensor node, the procedure is expensive and time consuming, and needs to be repeated after any modification of the node.

An alternative is to obtain the ATEX certification for a case that will contain the node, as shown in Figure 2. Such enclosures are available from many vendors and can be purchased for about 10 Euros. This is the industry's preferred way of obtaining ATEX compliance because it is cheaper and more flexible. Obviously there is a risk that the ATEX enclosure affects the communication links since the sensor node's antenna is inside the enclosure. Furthermore, the casing can shield the sensor node from the sun and weather conditions, as well as keeping the internal temperature higher than the external one.

Since the control and emergency systems require that data is transported timely and reliably, it is necessary that the communication protocols are capable of achieving the required communication performance even if the quality of the wireless channel is fluctuating. To enable an efficient design of such protocols, we study the range and the variance of these fluctuations with respect to the changes in ambient temperature and the use of ATEX casing.

III. Impact of Temperature on Communication

The outdoor deployment in the refinery is affected by frequent temperature changes and different weather conditions. Hardware components for outdoor deployments are usually designed for an operating temperature range from -40 °C to $+85$ °C. Temperature changes, however, cause a shift of the crystal frequency, increased thermal noise of the transceiver, and saturated amplifiers [4], resulting in degraded radio performance [5], [6].

A. Sentilla Tmote Sky Platform

To quantify the impact of temperature on a communication link, several experiments involving a couple of Tmote Sky nodes were carried out. The Sentilla Tmote Sky [7] uses the Chipcon 2420 radio chip [8] which operates at 2.4 GHz. The nodes run the Contiki operating system [9] with a customized application for the experiment. One node is used as a transmitter and the other node is a receiver. 256 packets, each with a 12-byte payload, are transmitted every 4 seconds. Nodes are placed at 3 meters distance, and their transmission power is kept at -3 dBm throughout the experiment. We use different 802.15.4 channels to make sure that specific interference on a channel is not biasing the data. The receiver logs the averages of the Received Signal Strength Indicator (RSSI), the Link Quality Indicator (LQI), the local temperature, and the sender's temperature which is contained in the received packets. The receiver also records the RSSI noise floor immediately after receiving each packet.

Different runs are carried out under different conditions: first both the sending node and the receiving node are exposed to an increase of temperature from between -15 and -3 °C to 53 °C in 90 minutes. The results of this experiment are shown in Figure 3. As we can see from the figure, the impact of temperature on the radio chip is considerable, and the higher the temperature is, the lower are the signal strength and the link quality. Figure 3 shows a signal strength drop of approximately 9 dBm. This is a substantial reduction, given that the typical range is between 0 and -100 dBm. Hence, high temperatures might lead to a loss of connectivity within the sensor network. Each point plotted is the result of an average operation over 256 packets. This enables us to monitor more precisely how the signal strength and link quality decrease, since the nominal RSSI and LQI are integer values.

Figure 4 shows that the RSSI noise floor decreases as well with temperature. This is an important observation because this value is often used by the medium access control (MAC) layer to determine if the channel is currently busy or not, and, as Figure 4 shows, also the noise floor is temperature dependent.

Under the same conditions, a second batch of experiments was carried out. Differently from the previous run, only the receiver node was exposed to a thermal variation from approximately -10 °C to 55 °C in 90 minutes. In this run we notice a drop of approximately 4 to 5 dBm in the RSSI when the temperature reaches the highest values. This variation is approximately 50% less than the one caused when both nodes are exposed to a thermal change. The RSSI noise floor decreases following the same pattern as in Figure 4.

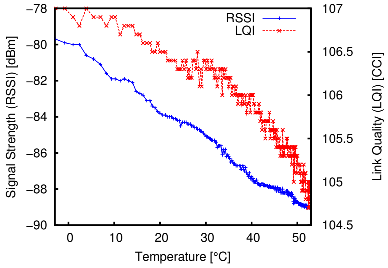


Fig. 3. Temperature impact on the RSSI and LQI indicators of the CC2420 radio chip when both sender and receiver nodes are affected by the thermal variation. Data is measured using the Sentilla Tmote Sky platform.

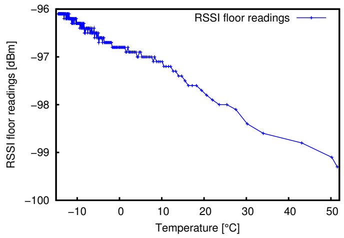


Fig. 4. In addition to the RSSI and LQI indicators, temperature has also an impact on the RSSI Noise floor readings of the CC2420 radio chip. Data is measured experimentally using the Sentilla Tmote Sky platform when both sender and receiver nodes are affected by the thermal variation.

We also carried out a third batch of runs in which only the sending node was exposed to a thermal change. Also under these conditions, we notice a drop of approximately 4 to 5 dBm in the RSSI when the temperature reaches the highest values.

Unlike the other two sets of experiments, however, no significant difference was noticed on the RSSI noise floor. We can thus infer that the drop shown in Figure 3 is the sum of two equal contributions: one due to the heated receiver, and one due to the heated sender.

B. Scatterweb MSB-430 Platform

In a second set of experiments we used the Scatterweb Modular Sensor Board [10] (MSB430) platform. This platform uses a CC1020 [11] radio chip running at 868 MHz. The experimental setup was similar to the setup used for the previously described experiment. However, only the RSSI and RSSI noise floor were recorded since the LQI is not available in the CC1020 radio.

As in the previous experiment, the impact on communication is highest when both sensor nodes are exposed to thermal change. Figure 5 shows the RSSI drop when temperature increases from -10 °C to 50 °C. The results show a similar dependence between temperature and RSSI as in the previous experiment: Figure 5 shows a signal strength drop of approximately 6 dBm over the investigated temperature range. Like with the Tmote Sky platform, the noise floor of the MSB430 platform also is affected by temperature, as shown in Figure 6.

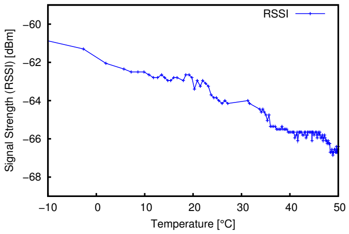


Fig. 5. Temperature impact on the RSSI of the CC1020 radio chip when both sender and receiver nodes are affected by the thermal variation. The data is measured using the Scatterweb MSB430 platform with an SMA antenna.

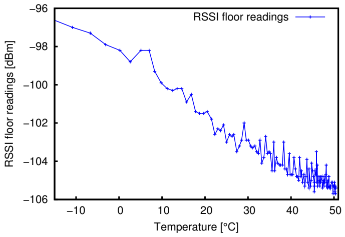


Fig. 6. In addition to the RSSI, temperature has also an impact on the RSSI Noise floor readings of the CC1020 radio chip. The data is measured using the Scatterweb MSB430 platform with an SMA antenna. Both sender and receiver nodes are affected by the thermal variation.

C. Discussion of the Obtained Results

From our experimental results it can be concluded that the observed temperature dependency exists on different platforms and for different radio frequencies. The observations are not antenna specific as the temperature impact is visible on both the Tmote Sky which has a built-in PCB antenna, and on the MSB430 which uses an external antenna connected via SMA. The effects are caused by the radio chip. More precisely, the components affected by temperature are the power amplifier of the transmitter and the LNA (that amplifies the RF signal from the antenna) of the receiver [4], [5].

If a temperature increase affects the power amplifier of the radio chip negatively, the signal strength of transmissions will decrease with the increasing temperature if the transmission power is constant. This partially explains why in our experiments the RSSI at the receiver decreases when the sender is warmed. At the same time, this means that a sensor node running at high temperatures needs a higher transmission power to obtain the same signal strength as is possible to obtain when transmitting at lower temperatures. This implies that also the transmission power is influenced by temperature.

In addition to this, the RSSI is further reduced when the receiver is exposed to high temperature, which is due to the lower LNA amplification. This impacts only the values of RSSI and RSSI noise that are returned by the chip because the environmental noise obviously does not decrease with temperature.

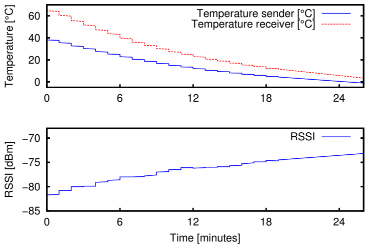


Fig. 7. Sensor nodes enclosed in an ATEX-compliant casing: a high temperature inside the case can be detrimental to low-power communication. Confirming the previous results, the RSSI decreases when the temperature increases. Temperature varies in the ATEX case of both sender and receiver.

IV. The ATEX Certification Requirement

In order to achieve ATEX compliance, sensor nodes can be enclosed in ATEX compliant casings. This procedure avoids costly certification procedures, especially for small modifications of the sensor node hardware.

A. ATEX Enclosures

When sensor nodes are enclosed in ATEX-compliant cases, the radio propagation might be affected by the casing. We carried out several experiments to evaluate if there is a decrease of performance when sensors are enclosed into ATEX cases. However, the results did not show a negative systematic trend of the RSSI when the nodes are inside the case. This applies at different distances and locations, both indoor and outdoor. The node's orientation, the deployment location, the presence of obstacles in the surroundings, and the environmental interference are the variables that affect the radio signal reception rather than the presence of the casing.

B. ATEX Enclosures and Temperature

The ATEX case does not have an impact on radio propagation and hence does not affect communication directly. However, the casing has an effect on the temperature of the sensor node and, thus, has an indirect impact on communication.

A high temperature inside the ATEX case can be detrimental to low-power communication, and the temperature effect is largest when the internal temperature of both the cases of sender and receiver is high. After carrying out the same set of experiments as in Section III-A, but enclosing both the sensor nodes in ATEX casings, we detected a rise in the RSSI of approximately 9 dBm, as shown in Figure 7.

The experimental results show again that temperature changes have a significant impact on communication. In order to ensure stable communication links, it might therefore be useful to avoid nodes exposed to direct sunlight. In the investigated refinery scenario this is possible since the deployment is highly controlled, and sensors are not deployed randomly.

The airtight ATEX casing creates a warming effect that increases the inner temperature. In our application, the temperature inside the ATEX cases may follow dangerous patterns with respect to our discussion in Section III, and might degrade

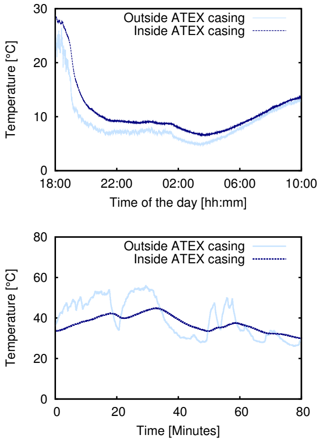


Fig. 8. Temperature registered on Tmote Sky nodes placed inside and outside ATEX-compliant enclosures at different hours of the day. The top figure shows how, during the night, the casing creates a warming effect on the sensor nodes, and the temperature inside the case is higher than outside. The bottom figure shows that when the sun shines directly on the sensor motes, the case shields the nodes, and slows down the inner increase of temperature.

the performance of the network or disrupt the connectivity between sensors. A high temperature–partly caused by the warming effect–can reduce the received signal strength.

For this reason we inspect the behavior of the temperature inside the ATEX casing with different weather conditions at different hours of the day. We use Contiki [9] and Sentilla Tmote Sky nodes equipped with Sensirion SHT11 temperature sensors [12] to perform such outdoor experiments. We compare the behavior of nodes enclosed in ATEX-compliant cases with nodes that are not enclosed. Figure 8 shows the temperature inside and outside the case at different times of the day. During nighttime, the airtight casing keeps the nodes at a higher temperature than the ones outside the case (top figure). During daytime, instead, when the sun shines directly on the sensors, the nodes outside the ATEX-compliant cases will be influenced faster, and the temperature will rise quickly (bottom figure). In other words, the case shields the sensor nodes and slows down the increase of temperature on the board, which helps to avoid sudden temperature changes.

This implies that sensor nodes may have enough time to modify the routing schemes before the temperature becomes too high. This is an important observation since the enclosure of sensor nodes in plastic cases is typically considered to bring disadvantages to the communication. In the case of the oil refinery, the indicatory system may switch the behavior from real-time data communication to data collection (i.e., waiting for the temperature to decrease again before transmitting). This will avoid retransmissions and a consequent waste of energy, and can be done since the data of the indicatory system is not time critical. The deployments for the control and emergency systems should instead be carried out so that even the highest temperature combined with the warming effect does not increase the latency of the real-time communication.

V. The Impact of Temperature on TX Power

Sections III and IV describe the influence of temperature and ATEX enclosures on transmission links in industrial outdoor deployments. The experimental results show that an increase in temperature leads to a reduction of the signal strength at the receiving side, due to the impact of temperature on the radio driver, and more precisely on the power amplifier of the transmitter [4], [5].

The impact of temperature on the power amplifier directly affects the strength of the outgoing radio signal: at higher temperatures the signal gets weaker. Therefore it can be expected that with an increase in temperature, a higher transmission power is required to maintain the same signal strength and thus to ensure successful data transmission.

We carried out several long-term outdoor experiments to investigate this effect. The aim was to determine the minimum transmission power level necessary to ensure successful data transmission between two Tmote Sky nodes. Given a pool of N packets, we define *the minimum power to reliably communicate* as the minimum power necessary to achieve 100% delivery, i.e. we expect exactly N received packets. Furthermore, we define *the minimum power to barely communicate* as the minimum power necessary to receive at least one packet, without caring about the actual delivery rate.

A. Experimental setup

We divide the deployed nodes in pairs consisting of a sending and a receiving node running the Contiki operating system [9]. The sender transmits a train of 15 packets with 12-byte payloads, starting with the highest transmission power available. Each packet contains a sequence number and the information about the transmission power used by the sender. The receiving node uses the same transmission power as advertised in the message to reply to the sender. The receiver sends an acknowledgment for every received packet, identified by its sequence number. If the sender receives at least one acknowledgment for the 15 packets sent, it will decrease the transmission power by one unit. We did not use a MAC protocol to organize channel access as we wanted to analyze only channel characteristics. We use static Tmote Sky nodes to run this experiment during different days and nights.

B. Transmission power levels in the CC2420 radio

The transmission power in the CC2420 radio driver can be set into 32 different values, ranging from roughly -55 dBm to 0 dBm through the PA_POWER register. Unfortunately, the CC2420 datasheet [4] documents only 8 discrete levels ranging from -25 dBm to 0 dBm, and the radio manufacturer confirms that the relationship between the register setting and the output power is not linear [13]. However, in order to compute the unknown values, estimations have been used, such as the cubic spline interpolation [14]. We measured the current consumption for all PA_POWER values experimentally using an oscilloscope. This is an important contribution of this paper, as information about the transmission power of the CC2420 does not exist [13], [14]. We measure the current consumption

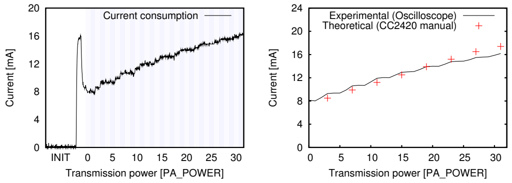


Fig. 9. Current consumption in the CC2420 radio chip. The left figure shows the current measured with the oscilloscope while increasing the PA_POWER level from 0 to 31. The first spike occurs because the radio is switched on and needs to be initialized (INIT). The right figure shows a comparison between the values provided by the CC2420 manual and the experimental results.

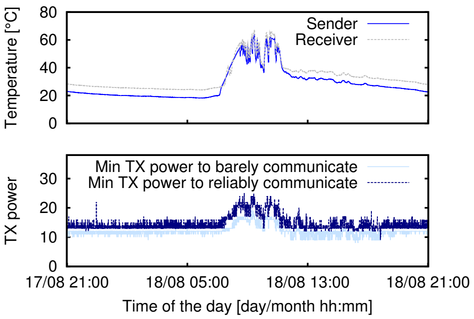


Fig. 10. Minimum transmission power required to communicate by two sensor nodes. During daytime the sun shines directly on the motes, and increases significantly the temperature on the board. With respect to nighttime operations, the sensor nodes require roughly 16% more energy for a successful transmission during the hottest time of the day. In this example, sensor nodes were deployed at a distance of around 7 meters and were exposed to sunlight from 8:00 to 11:00.

of the different PA_POWER levels using a Velleman PCSU 1000 oscilloscope [15] over a resistance of 100Ω. Figure 9 shows the characteristic of the current consumption of the Chipcon CC2420 radio that we measured experimentally. We confirm that the slope is not linear, which shows the importance of measuring the value for each power level. These values are used to calculate the current consumption for the PA_POWER values not specified in the CC2420 manual.

C. Experimental results

Our experimental results show that the minimum transmission power to communicate is considerably affected by temperature variations. This applies for temperature fluctuations between day and night and for changing weather conditions as well. All the results we obtained in our runs show a significant increase in the minimum transmission power, indicating that reducing the transmission power during the coldest time of the day or the year may help in saving energy.

Figure 10 shows a daily deployment in Germany during the summer, and we can see that when the sun shines on the sensor nodes, the temperature reaches up to 70 °C, thus 55 °C higher than during the night. Nodes are not exposed to wind, and they are placed approximately 7 meters away from each other. The high thermal variation causes an increase of the minimum transmission power to barely communicate from PA_POWER 11 to 17. At the same way, the minimum transmission power to reliably communicate increases from PA_POWER 13 to 22. According to our experimental results shown in Figure 9, the current consumption increases by 11.4% in the first case, and by 16.3% if we want to achieve a 100% delivery rate.

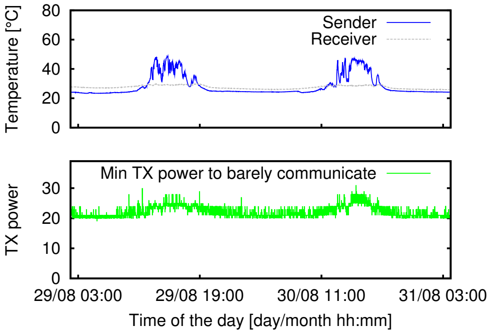


Fig. 11. Minimum transmission power required to communicate between two sensor nodes over multiple days. During daytime the sun shines directly on the sending node, and significantly increases the temperature on its board. With respect to nighttime operations, the sending node requires roughly 10% more energy for a successful transmission during the hottest time of the day. In this example, sensor nodes were at a distance of around 13 meters and–unlike in the previous experiment–only the sender was exposed to sunlight during the afternoon.

During this experiment, both sender and receiver nodes are affected by high temperature changes. This implies that in addition to the sender transmissions, also the acknowledgments sent from the receiver need a higher transmission power to reach their destination.

Figure 11 shows a deployment in Sweden during the end of August. The results are relative to a sunny weekend, where only the sending node is exposed to the sunlight. We can see that when the sun shines directly on the mote, the temperature increases up to 48 °C, thus 25 °C higher than during the night. Nodes are not exposed to wind, and they are approximately 13 meters far away from each other. The nodes are placed in such a position that they cannot achieve 100% delivery even with the highest transmission power available. However, we notice how the minimum transmission power to barely communicate increases from PA_POWER 20 to 28 when the temperature increases. According to our experimental results shown in Figure 9, the current consumption needed to barely communicate increases with 10.1%. This result confirms that even when the temperature variation is not as high as it was in the deployment in Germany, the impact of temperature is still considerable.

Another experiment was carried out in Sweden during the spring using different distances between the two Sky nodes (the distance was gradually increased from 50 cm to 20 m). The minimum transmission power was then compared when temperature in both nodes was 18 °C and 38 °C respectively. Figure 12 shows the results of the experiments, where PA_POWER represents the transmission power level used in the CC2420 radio. The plot shows the minimum transmission power to barely communicate, and confirms that

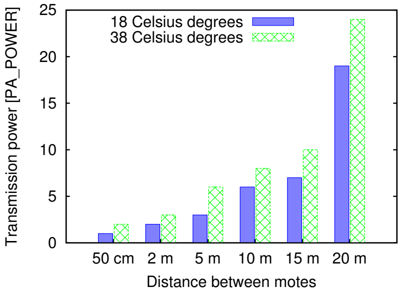


Fig. 12. Minimum transmission power required for a successful communication between two sensor nodes at different distances. Regardless of the distance between the motes, an higher temperature requires an higher transmission power to maintain a stable communication between nodes.

the temperature impact applies at all distances.

We further checked whether the nightly operations were requiring less energy because of the reduction in temperature or because of the minor environmental noise. During night there is not only a decrease in temperature but also a reduction of ambient noise because generally fewer electric devices are operating. We carried out many different experiments, and the ones shown in Figure 10 and 11 were explicitly chosen since they do not suffer from external interference. In the first plot, the sun is shining on the two motes only from 8:00 to 11:00, while the motes were in the shadow during the midday and the afternoon. We can clearly see that both temperature and transmission power decrease after 11:00, showing that the correlation is with the temperature rather than with the environmental noise. The second experiment is carried out in an office during weekends, so to avoid external interference.

In summary, our experimental results show that reducing the transmission power during nighttime and the coldest time of the year is a good practice that can save up to 16% of the energy consumption. Creating a control loop algorithm that adapts the transmission power to the temperature sensed by the sensor nodes may thus help to increase the overall network lifetime. Beside the temperature impact, it is also good to keep the transmission power as low as possible, because increasing the transmission power may result in more contention, although the link quality improves [16].

VI. Impact on Application Scenario

The influence of temperature variations on communication link quality must be taken into account when deploying a wireless sensor network in the application context outlined in Section II. In particular, the following aspects should be considered when designing and deploying a sensornet for the oil refinery context:

Deployment Time: The time chosen to deploy and test the equipment in the refinery is crucial. New devices within the refinery are typically deployed and tested during the evening or night when the refinery is at its quietest. In the south of Portugal, temperatures can vary in the summer between 35 °C during the day and 20 °C during night times. In addition, some of the nodes may be exposed to direct sunlight which will increase the temperature even further (see Figure 10). Thus, temperature variations between 18 °C and 38 °C as used to derive results shown in Figure 12 have to be expected. The graph shows that two devices can communicate over a greater distance when the environmental temperature is lower than at times of higher temperature. For example, a communication link configured with a transmission power level 3 to span a 5 meters distance at night will only be able to cover a distance of 2 meters during the day, which might result in a disconnected network. Hence, devices deployed and tested during the usual refinery maintenance period (which coincides with the coldest time) may not be able to communicate during daytime, when temperatures are higher. Therefore it is important that the communications are tested during the hottest times of the year.

Maintenance: Wireless sensor nodes within the refinery will be battery powered and therefore only have a finite lifetime. Continued operations can only be ensured when batteries are replaced before depletion. The cost of replacing batteries of 35,000 nodes within the refinery is very high and cannot be neglected. Maintenance personnel must be employed to ensure that batteries are replaced at the right time, which accounts for the largest part of the maintenance cost, while actual material cost for batteries is insignificant in comparison. Hence, it is important to achieve a long node lifetime to reduce the maintenance frequency. It is not advisable to use the maximum transmission power that a node provides. To conserve energy, the power should be set to the minimum required to bridge the required distance. Given the results shown in Figures 10, 11, and 12 the temperature dependency of the transmission power should be taken into account as well. Saving energy during nighttime and during the coldest seasons prolongs the battery duration, and therefore it is worth considering to adapt the transmission power to the ambient temperature.

Protocol Design: As pointed out in the previous paragraph it is necessary to take temperature into account also when deciding which transmission power should be used. Ideally, a node should adapt automatically to the proper transmission power setting. Generally, it is difficult to construct a stable adaptive algorithm if the temperature is fluctuating heavily over a short time span. However, as shown in Figure 8, the ATEX casing shields the sensor node from erratic temperature changes. Hence, we believe it is possible to devise a stable and efficient algorithm for transmission power adaptation, such as the one shown by Hackmann et al. [16].

VII. Related Work

Several researchers have shown that outdoor sensor networks are affected by weather conditions and temperature. Thelen et al. [17] described how radio waves propagate better under weather conditions with high humidity in their potato field deployment. The results of Anastasi et al. [18], Sun et al. [19], and Capsuto et al. [20] suggest that weather effects, specifically fog and rain, may have a severe impact on the transmission range of sensor nodes, in particular with respect to the packet reception rate. Boano et al. [6] quantified the impact on rain and fog with respect to the signal strength and the link quality under different platforms, showing that rainfall of less than 2-3 mm/hour has a negligible effect on the signal strength. When the rainfall is heavier, however, the connectivity might be disrupted.

Bannister et al. have shown that high temperatures negatively affect communication between sensor nodes [5]. In their deployment in the Sonoran Desert of the southwestern United States, the reduction of the signal strength was largest during the hottest time of the day. We quantify the impact of temperature also at lower temperatures, using different platforms and radio frequencies. We show that also the LQI, in addition to the RSSI, is affected. This is very important, since RSSI and LQI are used often to estimate the future packet reception rate of communication links [21], [22].

Unlike previous work, we show the influence that temperature has on the minimum transmission power necessary for communication between sensor nodes. We show that sensor networks operating at low temperatures can decrease their transmission power and save up to 16% energy, and thus increase their lifetime. To the best of our knowledge, this is a novel contribution. There are different protocols implemented to adapt the transmission power such as ATPC [23], but they adapt the transmission power based on neighbor status.

In order to obtain a high precision in our results, we measured the current consumption of all the 32 output power levels in the CC2420 radio chip. Our experimental results fill up the knowledge gap in the information provided by the manual [8] that is limiting researchers' work, as highlighted by Hauer et al. [13]. In this way, we evaluate the precise amount of energy saved, without resorting to empirical or statistical approaches as others have done [14] or using only the transmission channels for which the power consumption is provided by the manual as done by Hackman et al. [16]. We show how obtained transmission power is non-linear in relation to the configured power level (PA_POWER), and that a regression may not be the appropriate choice.

Most sensornets for industrial control and automation applications must comply with the ATEX directive 94/9/EC [1] for equipment and protective systems intended for use in potentially explosive atmospheres. To the best of our knowledge, there are no studies that assess if compliance with this standard has an impact on wireless sensor networks performance. Our measurements aim to close this knowledge gap.

VIII. Conclusions

In this article we investigated the temperature influence on low-power communications. For our case study, we used the deployment of an outdoor wireless sensor network in an oil refinery in which ATEX compliance is a necessity. Our experimental results show that temperature has a major effect on signal strength and link quality, and that operations at lower temperatures might require up to 16% less power to maintain a reliable communication. We have further explained how this affects the deployment and the design of the network in the refinery. We believe that the findings presented in this article can help to improve the design of wireless sensor network deployments for industrial process and control applications. Furthermore, the presented results can be used to construct energy-efficient protocols that adapt the transmission power to the measured ambient temperature in order to save energy and increase the lifetime of the system.

Acknowledgments

This work has been partially supported by the European Commission under FP7 with contracts FP7-ICT-224282 (GINSENG) and FP7-2007-2-224053 (CONET, the Cooperating Objects Network of Excellence). This work has been partially financed by SSF and VINNOVA, the Swedish Agency for Innovation Systems. This work has been also partially financed by the Cluster of Excellence 306/1 ”Inflammation at Interfaces” (Excellence Initiative, Germany, since 2006).

References

[1] "ATEX Guidelines on the Appl. of Dir. 94/9/EC: Equipment intended for use in Potentially Expl. Atmospheres. 3rd edition." Jun. 2009.

[2] "The GINSENG project: Performance control in industrial wireless sensor networks." [Online]. Available: http://www.ict-ginseng.eu/

[3] "GALP energy," Web page. [Online]. Available: http://www.galp.pt/

[4] *CC2400 datasheet - 2.4 GHz Low-Power RF Transceiver (Rev. 1.5)*, Chipcon AS, Mar. 2006.

[5] K. Bannister, G. Giorgetti, and S. Gupta, "Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization," in *Proc. of the 5th Workshop on Emb. Networked Sensors (HotEmNets)*, Charlottesville, Virginia, Jun. 2008.

[6] C. A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt, "Low-Power Radio Communication in Industrial Outdoor Deployments: The Impact of Weather Conditions and ATEX-compliance," in *Proceedings of the 1st Int. Conference on Sensor Networks Applications, Experimentation and Logistics (Sensappeal)*, Athens, Greece, Sep. 2009.

[7] *Tmote Sky datasheet*, Edition 1.04 ed., Moteiv Corporation, Nov. 2006.

[8] *CC2420 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B)*, Chipcon AS, Mar. 2007.

[9] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki: a lightweight and flexible operating system for tiny networked sensors," in *Proc. of the Workshop on Emb. Networked Sensors*, Tampa, Florida, Nov. 2004.

[10] *MSB: Modular Sensor Board datasheet*, ScatterWeb GmbH, Nov. 2007.

[11] *CC1020 datasheet - Low-Power RF Transceiver for Narrowband Systems (Rev. B)*, Chipcon AS, Jul. 2008.

[12] *SHT1x Humidity and Temperature Sensor datasheet*, Version 2.04 ed., Sensirion AG, May 2005.

[13] J.-H. Hauer, V. Handziski, and A. Wolisz, "Experimental Study of the Impact of WLAN Interference on IEEE 802.15.4 BAN," in *Proc. of 6th European Conf. on Wireless Sensor Networks*, Cork, Ireland, Feb. 2009.

[14] R. de Paz Alberola and D. Pesch, "AvroraZ: Extending Avrora with an IEEE 802.15.4 Compliant Radio Chip Model," in *Proc. of the 3rd 'PM2HW2N' Workshop*, Vancouver, Canada, Oct. 2008.

[15] Velleman, "Digital Storage Oscill. User Manual PCSU 1000," 2005.

[16] G. Hackmann, O. Chipara, and C. Lu, "Robust Topology Control for Indoor WSN," in *Proc. of the 6th Conf. on Networked Emb. Sensor Systems (SenSys)*, Raleigh, North Carolina, USA, Nov. 2008.

[17] J. Thelen, D. Goense, and K. Langendoen, "Radio wave propagation in potato fields," in *Proceedings of the 1st workshop on wireless network measurement (WiNMee'05)*, Riva del Garda, Italy, Apr. 2005.

[18] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proc. of the 7th MSWiM symposium*, Venice, Italy, Oct. 2004.

[19] J. Sun and R. C. Oliver, "An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks," in *Proc. of REALWSN'06*, Uppsala, Sweden, Jun. 2006.

[20] B. Capsuto and J. Frolik, "A system to monitor signal fade due to weather phenomena for outdoor sensor systems," in *Proc. of the 5th IPSN, Demo session*, Nashville, TN, USA, Apr. 2006.

[21] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *Proc. of the 3rd 'EmNetS' Workshop*, Cambridge, MA, USA, May 2006.

[22] M. Holland, R. Aures, and W. Heinzelman, "Experimental investigation of radio performance in WSN," in *Proc. of the Workshop on Wireless Mesh Networks (WiMesh)*, Reston, Virginia, Sep. 2006.

[23] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic, "ATPC: Adaptive Transmission Power Control for WSN," in *Proc. of the 4th SenSys Conference*, Boulder, Colorado, USA, Nov. 2006.

Paper C

C.A. Boano, T. Voigt, C. Noda, K. Römer, and M.A. Zúñiga. **JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation.** *In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* Chicago, IL, USA. April 2011. **Best Paper Nominee.**

Summary. This paper illustrates the design and implementation of JamLab, a low-cost infrastructure to augment existing sensornet testbeds with accurate interference generation. JamLab uses off-the-shelf sensor motes to record and playback interference in real-time within an existing WSN testbed, and this paper analyses how to get an accurate measurement and regeneration of interference despite the hardware limitations of common wireless sensor nodes. This paper further describes a practical procedure to augment an existing testbed infrastructure and to select and configure a fraction of the existing nodes as jammers. Finally, this paper evaluates experimentally the accuracy of the interference regenerated by JamLab with respect to time, space, and intensity, and shows how JamLab can be used to characterize the impact of interference on sensornet MAC protocols.

My contributions. I am the main author of this paper and was responsible for JamLab's design and implementation. I wrote the vast majority of the paper in collaboration and discussion with the co-authors, and carried out the majority of the experiments in the evaluation section. Thiemo Voigt and Marco Zuniga played a fundamental role in deriving the non-saturated and saturated model for Wi-Fi and Bluetooth (Section 5.1 and 5.2). Thiemo Voigt also helped in implementing the HandyMotes and in carrying out the experiments in Section 7.5. The saturation avoidance in high-frequency RSSI readings (Section 3.2) was conceived and implemented by Claro Noda. The actual name "JamLab" was conceived by Kay Römer, who also helped in conceiving JamLab's architecture and configuration. I presented the paper at IPSN'11.

ISBN: 978-1-61284-854-9.
DOI: 10.1109/IPSN.2011.5779101.

Link: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5779101.

- On the author's own home page;
- On the author's institutional repository;
- In any repository legally mandated by the agency funding the research on which the work is based.

JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation

Carlo Alberto Boano[†], Thiemo Voigt[‡], Claro Noda[¶], Kay Römer[†], and Marco Zúñiga[§]

[†]Institute of Computer Engineering
University of Lübeck, Germany
{cboano, roemer}@iti.uni-luebeck.de

[‡]Swedish Institute of Computer Science
Kista, Sweden
thiemo@sics.se

[¶]CISTER Research Unit
Polytechnic Institute of Porto, Portugal
cand@isep.ipp.pt

[§]Networked Embedded Systems Group
University of Duisburg-Essen, Germany
marco.zuniga@uni-due.de

ABSTRACT

Radio interference drastically affects the performance of sensornet communications, leading to packet loss and reduced energy-efficiency. As an increasing number of wireless devices operates on the same ISM frequencies, there is a strong need for understanding and debugging the performance of existing sensornet protocols under interference. Doing so requires a low-cost flexible testbed infrastructure that allows the repeatable generation of a wide range of interference patterns. Unfortunately, to date, existing sensornet testbeds lack such capabilities, and do not permit to study easily the coexistence problems between devices sharing the same frequencies. This paper addresses the current lack of such an infrastructure by using off-the-shelf sensor motes to record and playback interference patterns as well as to generate customizable and repeatable interference in real-time. We propose and develop JamLab: a low-cost infrastructure to augment existing sensornet testbeds with accurate interference generation while limiting the overhead to a simple upload of the appropriate software. We explain how we tackle the hardware limitations and get an accurate measurement and regeneration of interference, and we experimentally evaluate the accuracy of JamLab with respect to time, space, and intensity. We further use JamLab to characterize the impact of interference on sensornet MAC protocols.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids.

General Terms

Design, Experimentation, Measurement, Performance, Reliability.

Keywords

JamLab, HandyMote, Interference Generation, Wireless Sensor Networks, Augmenting Sensornet Testbeds.

IPSN'11, April 12–14, 2011, Chicago, Illinois.
Copyright 2011 ACM 978-1-4503-0512-9/11/04 ...$10.00.

1. INTRODUCTION

The reliability and robustness of sensornet communications are affected by radio interference. As an increasing number of standardized communication technologies operate in ISM bands, the congestion in the radio spectrum is inflating, and the quality of communications decreases. In safety-critical sensornet applications such as industrial automation and health care, in which the reliability and stability of communications are vital, radio interference represents a major challenge, as it leads to packet loss, high latencies, and reduced energy-efficiency due to retransmissions.

This issue is especially serious in the 2.4 GHz ISM band, as wireless sensor networks that operate at such frequencies must compete with the ongoing communications of WLAN, Bluetooth, and other IEEE 802.15.4 devices. Furthermore, sensornet communications in these frequencies can also be affected by several domestic appliances that are source of electromagnetic noise, such as microwave ovens, video-capture devices or baby monitors. This high number of different wireless devices sharing the same frequencies and space, raises the need for coexistence and interference mitigation techniques in 802.15.4-based sensor networks, as highlighted by previous studies [1, 2].

In particular, there is a strong need for understanding the performance of existing sensornet protocols under interference, as well as designing novel protocols that can deliver high and stable performance despite changing interference patterns. This, however, requires a proper testbed infrastructure where realistic interference patterns can be easily created in a precise and repeatable way. Unfortunately, existing sensornet testbeds lack such capabilities for interference generation, or they are limited to static WiFi access points randomly placed in the testbed [3], which does not enable the creation of a wide range of interference patterns in a repeatable way. Upgrading existing testbeds with additional heterogeneous devices in order to introduce interference sources is a costly, inflexible, labor-intensive, placement-dependent operation.

We therefore propose to augment existing sensornet testbeds with JamLab, a low-cost infrastructure for the creation of realistic and repeatable interference patterns. Such an infrastructure should support the recording and playback of interference traces in sensornet testbeds, as well as the customizable generation of typical interference patterns resulting from WiFi, Bluetooth, microwave ovens, or any other device operating in the frequency of interest.

To ensure a low-cost and hence widely applicable solution, we propose to use off-the-shelf motes. In this way, a fraction of the already deployed nodes of a testbed could be used for interference generation with the overhead limited to the simple uploading of the appropriate software. However, building such a low-cost solution is challenging due to the limitations of the available hardware. In order to obtain an accurate playback, the interference-pattern levels need to be measured precisely at a high sampling rate, so that also short interference patterns (e.g., resulting from WiFi traffic) can be recognized. We show in the paper how to obtain accurate readings of the RSSI noise floor while achieving a sampling frequency at up to 60 kHz. We show how at such high frequencies, many erroneous RSSI readings occur, and we correct such wrong readings by properly configuring the internal automatic gain control of the CC2420 radio. As a side effect, this technique also increases significantly the efficiency of Clear Channel Assessment (CCA) under interference. We exploit this approach to study the spectro-temporal characteristics of the most common interference sources such as WiFi, Bluetooth, and microwave ovens.

We further analyze and tackle the problem of (re)generating interference: the patterns have to be reproduced accurately in both frequency and time domains. This turned out to be hard to obtain, given the coarse output power levels available from the radio transceiver and the limited memory available on the motes. We show that to achieve an accurate regeneration, voluminous records of interference patterns need to be stored on the mote in real-time and later played back accordingly. Moreover, we provide precise and lightweight models of common interference sources in the 2.4 GHz ISM band to generate (emulate) realistic patterns.

Finally, the placement of the nodes inside the testbed is also critical. We study the implications in the spatial domain when measuring and generating interference in an indoor testbed and propose an optimal placement of the sensor nodes.

Our paper proceeds as follows: Section 2 describes the architecture of JamLab. Section 3 describes how we can use common sensor motes to accurately measure interference at high sampling rates avoiding erroneous RSSI readings. In Section 4 we show how to reproduce customized and repeatable interference patterns using sensor motes. In Section 5, we model several interference sources and show how we emulate specific interference patterns. We then discuss in Section 6 how to configure the testbed and the placement of nodes. We evaluate JamLab's accuracy in Section 7, and show how to practically augment an existing sensornet infrastructure. We further exploit JamLab to characterize the performance of sensornet protocols under emulated, but realistic interference. We review related work in Section 8 and conclude our paper in Section 9.

2. JAMLAB OVERVIEW

In sensor networks – especially in safety-critical applications with stringent quality-of-service requirements – robustness against interference is crucial. Interference may not only increase packet loss, the number of retransmissions, and therefore also power consumption, but timing constraints of the application may be violated and lead to failures. The observation that the environment has a profound impact on radio propagation has led to the excessive use of testbeds by the sensornet community, as simplified simulation models of radio propagation do not capture the complexity of the real world. The same holds true for interference: testbed infrastructures need to be augmented with means to generate realistic interference patterns in a repeatable manner to develop, test, and evaluate sensornet protocols and applications under interference.

With JamLab we propose a *low-cost* approach to augment *existing* testbeds with a way to generate *realistic* and *repeatable* interference patterns. The key idea behind JamLab is to use off-the-shelf motes to record and playback interference patterns instead of bringing WiFi access points, microwave ovens, or other equipment to the testbed. The latter approach is not only costly and hard to reproduce exactly by other researchers, but it is even difficult to exactly reproduce a given interference pattern with the same appliance. For example, the sequence and timing of the WiFi frames generated by a file download may differ between repeated trials due to TCP adaptation mechanisms (e.g., timeouts, window sizes). Furthermore, every device used to generate interference in the testbed needs to be programmed remotely. Programming several heterogeneous devices such as WiFi access points or microwave ovens would create a significant overhead, whereas using JamLab the installation overhead is minimal.

Indeed, with JamLab, either a fraction of the existing nodes in a testbed are used to record and playback interference patterns, or a few additional motes are placed in the testbed area. We call those motes used for interference generation *HandyMotes*. The HandyMotes support two modes of operation: *emulation*, where a simplified model is used to generate interference patterns that resemble those generated by a specific appliance (such as a WiFi device or a microwave oven); and *regeneration*, where each HandyMote autonomously samples the actual interference, compresses and stores it locally, and regenerates the recorded patterns later. The latter mode is especially useful to record realistic interference patterns in a crowded shopping center or on a lively street by placing a few HandyMotes to record interference, and bringing them to the testbed to playback the recorded traces there.

One fundamental challenge results from the fact that the maximum RF output power of motes (0 dBm) is typically much smaller than the RF output of other typical interference sources (25 and 60 dBm for WiFi and microwave ovens, respectively). Therefore, a WiFi transmitter or a microwave oven may disturb sensornet communications over much larger distances than a HandyMote can. We address this issue by subdividing the testbed area into cells as depicted in Figure 1, such that a HandyMote placed at the center of the cell can interfere with all testbed motes contained in the cell, but the interference with motes outsides of the cell is minimized. This requires a careful placement or selection of HandyMotes and control of their RF output power. We investigate this issue and propose a procedure for HandyMote placement and power control in Section 6. Note that there is a tradeoff between the realism of the generated interference patterns and the number of HandyMotes: the more cells, the more accurate is the spatial distribution of interference, but the more HandyMotes are required.

Another challenge is that many interference sources emit wideband signals, i.e., they interfere with many 802.15.4 communication channels at the same time. In contrast, a mote can only transmit on a single channel at a time. Fortunately, most existing sensornet protocols use only a single channel. However, there is a trend to use multiple 802.15.4 channels at different nodes to increase robustness and bandwidth. Our approach to deal with this issue is to place multiple HandyMotes in each cell, each one interfering on one 802.15.4 channel as detailed in Section 4.3. The use of Software Defined Radio (SDR) techniques using USRP devices would provide more accurate jamming signals on a wider bandwidth, but their high cost represents a sizeable limitation. To synchronize the generation of interference patterns within the HandyMotes in one cell and across cells, we need time synchronization, and we propose to use the testbed infrastructure (i.e., wired backchannels) to send synchronization signals to the HandyMotes.

Due to the constrained resources of a mote, also the accurate recording and playback of interference represent a challenge. To capture short interference patterns such as those generated by WiFi beacons, we need high sampling rates with low jitter, which requires data compression due to the limited amount of available memory. Our solutions to these problems are described in Sec-

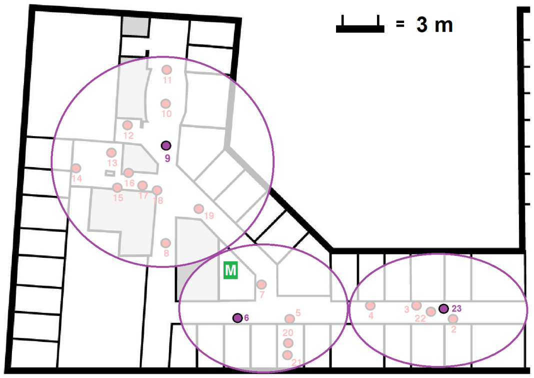


Figure 1: Testbed augmented with JamLab. Nodes 6, 9, and 23 are selected as HandyMotes, and take care of interference (re)generation in their cell.

tion 4.1. The accurate measurement of the interfering signal strength turned out to be a challenge in itself due to the gain control in the radio. Our solution to this problem is detailed in Section 3.

For the playback of recorded interference traces, normal packet transmissions are not appropriate, as this would offer only limited control over the exact timing of the transmitted signals. Therefore, we use special test modes of 802.15.4 radios to generate modulated or unmodulated carrier signals as detailed in Section 4.2. Those radios offer only a small number of discrete output power levels. While this can be exploited for compression of recorded traces, it limits the control over the generated interfering signal strength. However, we are primarily interested in *binary interference* generation, where a HandyMote either blocks the communication of the motes in its cell by emitting a strong-enough interference signal, or by not interfering at all. Nevertheless, HandyMote also supports the generation of a small number of output power levels as supported by the radio hardware, as discussed in Section 4.1.

JamLab has been designed specifically for the Texas Instruments CC2420 radio [4], and tested on several sensor motes such as Maxfor MTM-CM5000MSP, Crossbow TelosB, and Sentilla JCreate, but the framework can be applied to any sensornet platform. Based on the analysis of the datasheets, the Handymotes should be easily ported to similar radios such as the Ember EM2420 transceiver, and to newer radios such as the CC2520. We develop the HandyMotes based on Contiki, a lightweight and flexible operating system for tiny networked sensors [5].

3. MEASURING INTERFERENCE ACCURATELY USING MOTES

Measuring interference accurately on a mote is a key functionality, both for recording and later playback of interference, as well as for acquiring a deep understanding of common interference sources such as WiFi or Bluetooth. We describe in this section the techniques we used in order to let a common sensor mote measure the interference accurately at a sufficiently high sampling rate.

3.1 Measuring at High Sampling Rates

Link quality indicators such as RSSI and LQI provide an indication of the signal strength and quality, but only upon the reception of a packet. The only feasible way to assess the interference status is hence the continuous measurement of the RSSI noise floor, i.e., the RSSI in absence of packet transmissions.

In order to retrieve the spectro-temporal characteristics of different interference sources, we improve existing Contiki tools [6] and develop two applications that scan the 2.4 GHz frequency spectrum by reading the RSSI noise floor from the CC2420 radio transceiver:

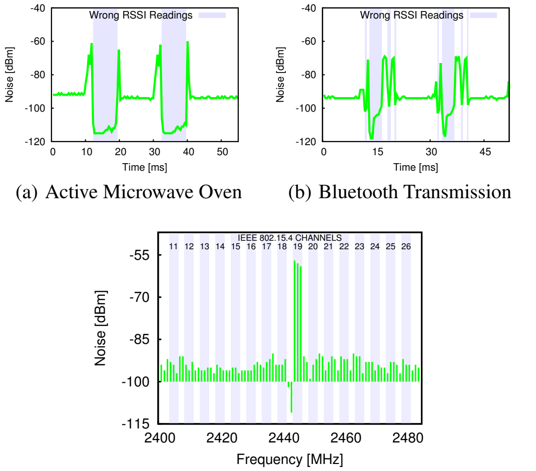
- The *time scanner* scans a single predefined IEEE 802.15.4 channel at its middle frequency with a very high sampling rate, and returns the RSSI noise floor readings over time;
- The *frequency scanner* scans sequentially the whole 2.4 GHz spectrum by switching between all 802.15.4 channels.

A first requirement of both scanners is to achieve a high sampling rate, given that we need to detect short transmissions periods. After boosting the CPU speed, optimizing the SPI operations, as well as buffering and compressing the RSSI noise floor readings using Run-Length Encoding (RLE), we reached a maximum sampling rate of approximately 60.5 kHz when sampling a single channel with the *time scanner*. The highest sampling frequency reachable by the *frequency scanner* is instead 3.4 kHz, since it is constrained by the settling time of the radio when switching channels. Hence, the limitations of low-power radios do not permit to achieve a sampling rate sufficiently high to capture all WiFi transmissions, as the maximum speed of 802.11b/g/n standards is 11, 54, and 150 Mbit/s, respectively. The minimum size of a WiFi packet is 38 bytes (ACK and CTS frames), which would make a resolution of 60 kHz sufficient to detect all 802.11b frames, but not all 802.11g/n frames. As most WiFi frames are data frames and typically contain higher layer headers, one can sample at 60 kHz frames with TCP/IP headers having a payload size higher than 27 and 227 bytes for 802.11g/n, respectively. Despite the use of large PDUs to reduce preamble overhead [7], this resolution does not guarantee to capture all the VoIP traffic over 802.11g/n [8].

Another requirement for the scanners is to accurately measure the strength of the ongoing interference in the radio spectrum by means of precise RSSI noise floor readings. The CC2420 radio specifies an accuracy of ±6 dBm, and a linearity of ±3 dB in the dynamic range $[-100, 0]$ dBm. Such accuracy and linearity has so far been acknowledged by the research community as enough to carry out operations such as Clear Channel Assessment (CCA) and low-power channel sampling for activity recognition [9]. However, our experiments show that the RSSI noise floor readings captured at high sampling rate suffer of a systematic problem in three specific scenarios, namely: (i) when a narrow unmodulated carrier is transmitted, (ii) when microwave ovens are switched on, and (iii) in the presence of Bluetooth transmissions. In these scenarios, the CC2420 radio often returns RSSI values that are significantly below the supported range and the sensitivity threshold, e.g., -110 or -115 dBm. Figure 2 reports examples of such wrong readings, which represent an important problem, since they also impact the correct functioning of CCA in the presence of narrow-band signals, as shown in Figure 2(c). Our investigation also shows that the same problems applies to other sensornet platforms employing similar versions of the chip, such as the Ember EM2420 transceiver. We experimentally identified that the problem is due to the saturation of the Intermediate Frequency (IF) amplifier chain: we have observed that maximum gain is used in the Variable Gain Amplifier (VGA) when the incorrect RSSI readings occur.

3.2 Avoiding Saturation in RSSI Readings

The reason of this saturation problem can be found in the radio demodulation chain. The CC2420 chip implements part of the IF filtering in analog domain and further filtering is later performed in digital domain. It employs an Automatic Gain Control (AGC) loop to maintain the signal amplitude close to a certain target value that guarantees the correct operation of the Analog-to-Digital Converter



(a) Active Microwave Oven (b) Bluetooth Transmission

(c) Unmodulated Carrier

Figure 2: Examples of wrong RSSI readings: several values are significantly below the sensitivity threshold of -100 dBm due to receiver saturation. This error is caused by an incorrect operation of the AGC loop in presence of narrow-band signals.

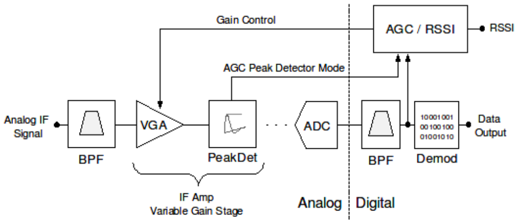


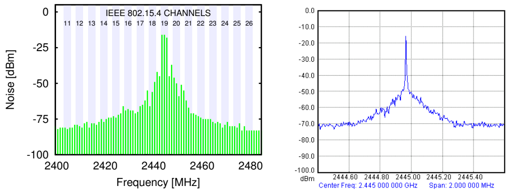
Figure 3: Simplified diagram of the CC2420 AGC loop.

(ADC). More specifically, the signal is maintained within the ADC dynamic range, despite large variations in the input signal from the antenna. For this purpose, the AGC loop uses a digital sample of the final IF signal amplitude and adjusts the gain of the VGA stage accordingly (see Figure 3). If a narrowband signal is present near the cut-off frequency of the combined IF chain, the resulting sampled signal amplitude may be remarkably lower than the partially unfiltered one at the ADC, as a consequence of the digital filtering. Since the AGC uses the final value to set the gain of the amplifier chain, there is no guarantee that the ADC is not saturating. In the event of ADC saturation, the receiver is no longer linear and the RSSI values are incorrect.

To linearize the radio response for an arbitrary noise signal and hence avoid wrong RSSI readings, we activate the peak detectors in-between the amplifier stages so that their output is used by the AGC algorithm to compute the required gain. The latter is attained with VGA stages and the system switches in and out fixed gain stages as needed. In the CC2420, the peak detectors are controlled by the *AGCTST1* register, and can be configured as follows:

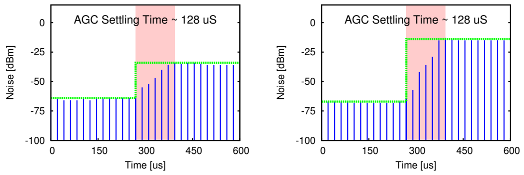
```
unsigned temp;
CC2420_READ_REG(CC2420_AGCTST1, temp);
CC2420_WRITE_REG(CC2420_AGCTST1,
(temp + (1 << 8) + (1 << 13)));
```

The register also includes flag bits to activate peak detectors among fixed gain stages in the IF chain and at the ADC itself [4].



(a) Sensor Mote (b) Anritsu MS2711D

Figure 4: Single tone excitation obtained running the *frequency scanner* operating across the band (a), and in the Anritsu Spectrum Analyzer, with a frequency span of 2 MHz (b). Notice the correct readings despite the very narrow pulse used, as compared to Figure 2(c).



(a) -25 dBm (b) 0 dBm

Figure 5: Evolution of RSSI readings over time to different RF tone step signals. The accuracy of our RSSI scanner is high enough to show the moving average used by the CC2420 to compute the RSSI over the last 8 received symbols.

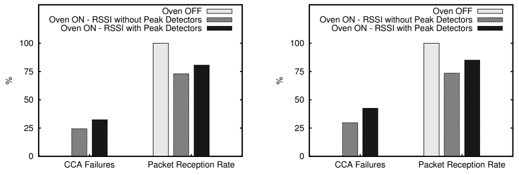
3.3 Validation of the Experimental Setup

We validate our RSSI noise floor measurements both in time and frequency with the help of a professional Anritsu MS2711D spectrum analyzer [10]. In these experiments, we connect the RF ports of the transceivers or the analyzer directly via a 50 Ohms matched impedance RF pigtail. This isolates the signals of interest from external noise sources and eliminates the medium pathloss, so that the amplitude of the tone and the spectral footprint can be compared.

Firstly, we verify the correctness of the *frequency scanner* readings, using the unmodulated test signal available in the CC2420 radio. In order to do this, we program another mote to transmit an unmodulated tone tuned at 2445 MHz, the center of IEEE-802.15.4 channel 19, at maximum power. Figure 4(a) shows the correct operation of the receiver and the linearized IF amplifier chain while scanning the RSSI values across the band using the peak detectors. The same test signal can be seen in the spectrum analyzer (Figure 4(b)). This worst case scenario shows that we have linearized the receiver, thus avoiding wrong RSSI noise floor readings.

Secondly, we measure the evolution of the RSSI readings over time to an RF tone step signal in order to evaluate the accuracy with which we can effectively measure RSSI values. We use our *time scanner* with two different power levels (-25 and 0 dBm), and obtain the results shown in Figure 5. The frequency of the scanner is sufficiently high to show how the CC2420 internally averages the RSSI over the last 8 received symbols, or 128 μs, as defined by the IEEE 802.15.4 standard. Such settling time is shown to be independent of the height of the step signal.

Impact on Clear Channel Assessment (CCA). Activating the peak detectors in-between the amplifier stages also improves the reliability of the CCA operation commonly used in MAC protocols [9]. Due to wrong RSSI readings, the CCA returns a clear channel when a narrow unmodulated signal is transmitted. As a result of this, the application would generate a transmission that is very likely to fail, thus wasting some of the limited energy budget.



(a) Channel 23

(b) Channel 25

Figure 6: Avoiding wrong RSSI readings improves the CCA accuracy and packet reception rate under interference.

A typical example of this would happen when transmitting packets in presence of an active Bluetooth device or a microwave oven in the neighborhood. Our approach significantly improves the CCA accuracy, leading to a higher Packet Reception Rate (PRR).

Figure 6 shows the amount of "channel busy" outcomes of CCA before and after activating the peak detectors. The absolute gain in terms of PRR depends on the microwave oven model, on the channel of interest, and on the data rate. We experimentally collect data at the receiver side of a couple of sensor nodes communicating periodically at a rate of 128 packets/second in presence of an active Lunik 200 microwave oven in the neighborhood. The nodes are placed at 1 meter distance and use a transmission power of -25 dBm. As shown in Figure 6, the PRR increases by up to 12% when activating the peak detectors and avoiding wrong RSSI readings.

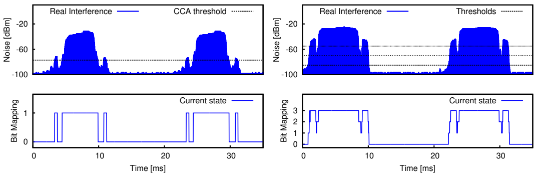
4. (RE)GENERATING INTERFERENCE

With the techniques to accurately measure interference introduced in the previous section, we can now proceed to record and replay those patterns. We describe first how to compress and store traces on motes and then how to playback those recordings.

4.1 Recording Interference Traces

When used in *regeneration* mode, HandyMote records interference traces that are later played back accordingly. Those traces can be either stored on the mote in RAM or Flash memory, or – if the HandyMote is connected to a testbed during recording – can be streamed over a wired backchannel to a base station. In any case, the data rate of 480 kbps generated by sampling RSSI with a resolution of 8 bits to hold values between 0 and -100 dBm at 60 kHz is too high to store it directly in memory or to stream it over the backchannel. The very efficient Coffee Flash file system supports a peak write bandwidth of only 376 kbps [11], the MSP430 UART supports a maximum data rate of 460 kbps for writing to the USB backchannel, and the limited 4 kB RAM of the MSP430 could just record a trace of less than 70 milliseconds duration.

While we need a high compression ratio, the compression method has to be efficient enough to allow sampling of RSSI at 60 kHz. Therefore, we use a simple Run-Length Encoding strategy and a quantization of the samples to a few bits per sample. We store a stream of pairs (v, o), where v is a sample and o is the number of consecutive occurrences of this sample. This method is very effective, as RSSI values typically change slowly over time. The quantization is justified by the fact that the CC2420 only supports 11 distinct output power levels in the range [-55,0] dBm by setting the PA_POWER register to the values we derived and listed in Table 4.1. To obtain the highest possible output resolution, four bits per sample with an appropriate non-linear quantization are hence sufficient. For example, for two-bit resolution one can use thresholds -55, -70, and -80 dBm (or register values 31, 7, and 3) with a spacing of 15 and 10 dBm, respectively, for quantizing the RSSI range into four regions.



(a) 1-bit precision

(b) 2-bit precision

Figure 7: Encoding techniques to save memory resources.

Figure 7(b) shows how original RSSI readings (top) are mapped into 2 bits (bottom): the two-bit quantization of a 35 ms interference recording reduces the amount of data from 2076 Bytes to 84 bytes – a compression ratio of $\frac{1}{25}$. A single bit per sample is enough to support binary interference regeneration. This corresponds to the outcome of a continuous CCA operation, in which the outcome busy/idle channel is mapped to a binary number [12]. Figure 7(a) shows the outcome of a one-bit quantization of 35 ms of interference. The amount of data is reduced from 2076 Bytes to 20 Bytes – a compression ratio of less than $\frac{1}{100}$. This reduces the raw data rate of 480 kbps to less than 5 kbps (depending of course on the values of the raw samples), a data rate that can be handled by Flash and USB, and allowing us to store several seconds of recording in RAM. In our current implementation, we store traces in RAM.

Recording interference traces is energy demanding, as both CPU and radio need to be constantly active while scanning the radio medium. Using software-based on-line energy estimation [13], we obtain an average power consumption of 65.4 mW for Tmote Sky motes, which allows for a lifetime up to 4 days when powered using primary AA batteries.

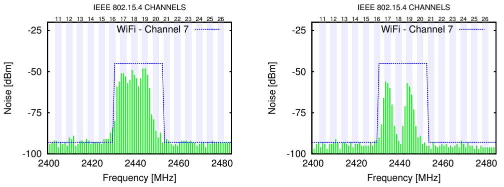
PA_POW.	dBm	PA_POW.	dBm	PA_POW.	dBm
31	0	15	-7	2	-45
27	-1	11	-10	1	-50
23	-3	7	-15	0	-55
19	-5	3	-25	-	-

Table 1: Discrete output power levels of the CC2420 radio.

4.2 Generating Interference Patterns

We have recently shown how the CC2420 test modes can be used to generate controllable and repeatable interference [14, 15] by transmitting a modulated or unmodulated carrier signal that is stable over time. This approach is superior to common jamming techniques based on packet transmissions, as the emitted carrier signal is independent from packet sizes and inter-packet times.

In order to generate an interference pattern, the interferer has to be enabled and disabled and its output power has to be set according to the compressed recorded trace in regeneration mode or according to the output of models in emulation mode, as described in Section 5. When enabling the transmitter using the STXON command, the radio oscillator first has to stabilize before a transmission is possible, resulting in a latency of 192μs or a maximum playback frequency of only 5 kHz. Therefore, we leave the transmitter on and just change the output power level to 0 (or -55 dBm) instead of disabling the transmitter. At level 0 the RF output power is so small that even a receiver at a distance of only few centimeters can hardly notice the signal. The advantage of this approach is that the latency for changing the output power is dominated by the SPI access time. We optimized the SPI driver in Contiki, resulting in a latency of only few microseconds – allowing us to to playback at the same frequency of 60 kHz that was also used during recording.



(a) HandyMote (ch. 17,18,19) (b) HandyMote (ch. 17,19 only)

Figure 8: Compared to wideband interferers, the HandyMotes jam only selected channels, preserving connectivity in others.

Besides the sampling and playback rate, also the jitter during playback of the individual samples needs to be minimized in order to ensure an accurate reconstruction. At 60 kHz, the playback time between two consecutive samples is just 17μs, hence the duration of the execution of a sequence of microcontroller instructions is no longer negligible. In particular, different execution paths in the program to uncompress samples in regeneration mode lead to different execution times and jitter. Therefore, we add NOP instructions to make all execution paths equally long.

4.3 Multiple Channels

Sensor motes are designed to transmit in only one of the 16 IEEE 802.15.4 channels. As most of the existing sensornet protocols only use a single channel, a single HandyMote is sufficient to interfere with this channel. However, there is an increasing trend to use multiple channels in order to increase robustness and bandwidth. In this case, we use multiple HandyMotes, each one interfering with one channel, as depicted in Figure 8(a). Using this approach, the interfered channels can be carefully selected as shown in Figure 8(b), and one can therefore avoid to jam other sensor networks installed in the same building. In contrast, a wideband interference source such as WiFi always jams at least 4 adjacent 802.15.4 channels.

For the synchronization we assume that all HandyMotes are connected to a basestation computer via USB cables and hubs as it is common in existing testbeds, such as TWIST and MoteLab. The synchronization algorithm is inspired by [16] and works as follows. For every HandyMote, the basestation sends a stream of N packets to the HandyMote. Just before sending the packet, the basestation reads its clock and includes the send timestamp in the message. Upon reception, the HandyMote reads its local clock to obtain the receive timestamp. It then computes the difference between send and receive timestamps for each of the N messages, chooses the message with the smallest difference (this is the message that had the smallest latency), and corrects its local clock by this difference. The synchronization error then depends on the variability of the observed minimum latencies across different HandyMotes. As the minimum latency is rather stable across different HandyMotes, the HandyMotes will be synchronized accurately among each other (but not necessarily with the basestation).

To estimate the accuracy of synchronization, we measure the clock offset between two HandyMotes synchronized in that way by having them trigger one of their digital output pins when their synchronized clock reaches a given time. By connecting the output pins of the two HandyMotes to an oscilloscope we measure an average offset of 8.44μs with a standard devitation of 6.94μs for a stream of N=10 packets. Increasing N further does not substantially reduce error. This accuracy is enough to ensure synchronization of motes playing back interference at 60 kHz.

5. MODELING INTERFERENCE SOURCES

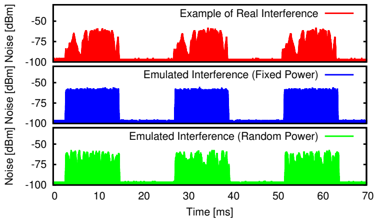


Figure 9: Emulation of microwave oven interference (top) with fixed (middle) and random power (bottom).

In this section, we describe how we can use an HandyMote to emulate three major sources of external interference on the 2.4 GHz ISM band: WiFi and Bluetooth devices, as well as microwave ovens. We present models that capture the temporal characteristics of these interference sources. A key requirement is the simplicity and efficiency of models, as they need to be executed in real-time on the HandyMotes to generate interference. We are not concerned about the intensity of the generated interference, since when running a HandyMote in *Emulation Mode*, we can decide to adjust the output power of the CC2420 according to different schemes. For example, the output power can be kept fixed or chosen randomly, as shown in Figure 9 (emulation of the interference generated by a Whirlpool M440 microwave oven).

5.1 WiFi Emulation

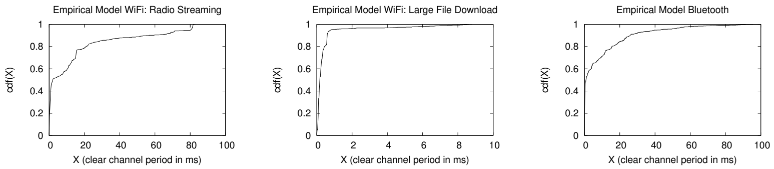
Modeling Wi-Fi traffic is challenging, as it depends on several factors such as the number of active users, their activities, the protocols they use (UDP or TCP), the traffic conditions in the backbone, etc. Under some reasonable assumptions, several theoretical studies have analyzed the performance of 802.11 [17, 18, 19, 20]. However, based on the empirical data we collected, we observed that the models for saturated sources provide a better approximation than the models for non-saturated sources (saturated sources always have data to send). Hence, in order to re-create a realistic representation of interference patterns, we implement an analytical model for saturated traffic sources, and for non-saturated traffic we derive models from empirical data.

Non-Saturated Traffic: Empirical Model. The empirical model for non-saturated traffic is obtained in the following way. Let us denote a random variable X as the *clear* time between two consecutive *busy* times. We obtain the probability mass function $p(x) = P_r\{X = x\}$ from the empirical sampling of the channel, where x is the time in number of slots (each slot is 1 ms). The length of the *busy* times is represented by the transmission delay of packets, which is a rather deterministic variable (for a fixed packet size). Following the methodology described on the previous paragraph, we obtained the $p(x)$ for the scenarios presented on Table 2. Figures 10(a) and 10(b) show the probability mass function $p(x)$ for two WiFi scenarios: an audio-stream application and the download of a large file.

Scenario	Users	Scenario	Users
Radio Str.	1	Video Str.	1
File Transfer	1	File + Radio	1

Table 2: Scenarios.

Saturated Traffic: Analytical Model. There exist several analytical models for the Distributed Coordination Function (DCF) mode of 802.11. Among them, the model proposed by Bianchi [18] has been one the most influential. Bianchi modeled the DCF mode of 802.11 as a discrete Markov process, where the back-off and retransmission mechanisms are represented as discrete states. Based on this model, Garetto and Chiasserini [19] developed a simpler Markov process by merging back-off states. For details, we re-



(a)

(b)

(c)

Figure 10: Empirical Models for WiFi and Bluetooth.

fer the reader to their paper [19]. In our work, we use Garetto and Chiasserini's model to emulate WiFi interference for saturated sources: whenever there are transmissions of frames in the model, the HandyMote activates the carrier.

5.2 Bluetooth Emulation

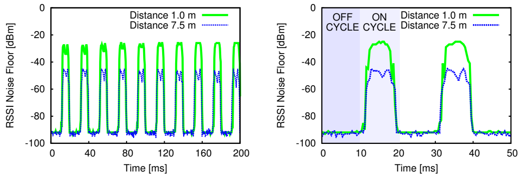
IEEE 802.15.1, better known as Bluetooth, specifies 79 channels, spaced 1 MHz, in the unlicensed 2.4 GHz ISM band. Bluetooth stack implementations apply an Adaptive Frequency Hopping (AFH) mechanism to combat interference, which does not permit to anticipate the frequency at which the interference will take place. Bluetooth hops 1600 times/sec., which means that it remains in a channel for at most 625μs. Note that Bluetooth channels are 1 MHz-spaced, while the resolution of our scanner is 2 MHz, which implies that consecutive time slots may eventually coincide within this frequency window and result in a larger interference period. We model Bluetooth using the same method as for non-saturated traffic in WiFi, that is, we obtain the probability density function $p(x)$ for the *clear* periods of the channel, and the transmission time of Bluetooth packets for the *busy* periods. Figure 10(c) shows the probability mass function $p(x)$ for the Bluetooth scenario. The Adaptive Frequency Hopping characteristic of Bluetooth leads to a smoother *cdf* curve compared to WiFi, because the *clear* periods are independent of the application run.

5.3 Microwave Oven Emulation

Microwave ovens are a kitchen appliance used to cook or warm food by passing non-ionizing microwave radiations to heat water and other polarized molecules within the food, usually at a frequency of 2.45 GHz. Therefore, they are a potential source of interference for sensornets operating in the 2.4 GHz spectrum.

The detailed characteristics of the interference patterns emitted by domestic microwave ovens depend on the model; nevertheless they all present the same basic properties. Firstly, on a spectral basis, our experiments show that microwave ovens tend to interfere all the 802.15.4 channels, with a higher impact on channels 20-26. It is not possible to state with certainty which channel will be mostly affected, as our experiments confirm that the peak frequency of the ovens depends on multiple factors, including the oven content, the amount of water in the food, and the position within the oven, as all these parameters affect the temperature of the magnetron [21]. Secondly, on a temporal basis, the generated noise is rigorously periodic. Figure 11 shows the temporal pattern of the interference caused by a Lunik 200 microwave oven retrieved experimentally. In one period of approximately 20 ms, there is an 'on' and 'off' cycle, whose duration is roughly 10 ms each. This matches the observations in [22], confirming the correctness of our results.

For all the above reasons, microwave oven interference is the simplest dynamic to model, as it follows a deterministic on/off sequence. Defining the period of the signal τ, the duty cycle λ (fraction of time the oven is 'on'), and hardcoding these two parameters into the HandyMote, we can generate interference patterns such as the ones shown in Figure 9.



(a) Zoom out

(b) Zoom in

Figure 11: Temporal characteristics of the interference caused by microwave ovens. The ovens emit frequencies with a periodic pattern with period $T \approx 20$ ms.

6. TESTBED CONFIGURATION

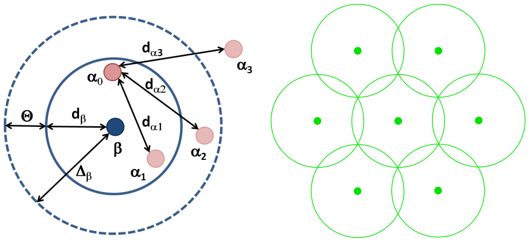
As outlined in Section 2, we partition the area of a testbed into different cells to deal with the limited RF output power of the HandyMotes compared to interference sources such as WiFi or microwave ovens. In this section we explain how to configure the testbed, i.e., how to place the HandyMotes and how to control their RF output power level such that the motes in the testbed are partitioned. This implies that every mote should be covered by a cell and cross-talk between neighboring cells should be minimized (i.e., a HandyMote does not interfere with motes outside of its cell).

6.1 Coverage and Cross-Talk

A key issue we need to understand is under which conditions the packet reception of a mote is actually affected by an interference signal generated by a HandyMote. The impact of interference on reception in the CC2420 radio is closely dependent on the modulation scheme used, namely OQPSK (Offset Quadrature Phase Shift Keying) and DSSS (Direct Sequence Spread Spectrum). With these modulation schemes, the interference signals generated by two HandyMotes do not simply "add up" at the receiver as it would be the case for ASK (Amplitude Shift Keying) used in older radios, but the receiver will pick the stronger of the two signals if their strength differs by a certain minimum. This is called co-channel rejection: according to [4], the CC2420 is able to receive a signal at -82 dBm if the second signal is at least 3 dB weaker.

In order to enable a HandyMote to interfere with the motes in its cell, we therefore need to make sure that a mote belonging to the cell will receive interference signals from that HandyMote with a signal strength at least 3 dB higher than the maximum strength of other signals that mote may receive. To minimize cross-talk between neighboring cells, we need to make sure that motes outside of the cell will receive that interference signal with a signal strength that is at least 3 dB weaker than the minimum strength of other signals that this mote may receive. Finally, we need to make sure that all testbed motes are covered by the cells.

In practice, an ideal configuration without cross-talk and with complete coverage typically does not exist. Also, due to environmental dynamics, the amount of cross-talk and coverage may vary over time. We can only try to find a configuration that maximizes coverage and minimizes cross-talk. Note that there is a tradeoff



(a) (b)

Figure 12: JamLab's division in cells.

between the size of the cells and the accuracy of the spatial distribution of generated interference: the smaller the cells, the higher is the spatial sampling resolution and the smaller are the cross-talk regions. However, smaller cells also implies that more HandyMotes are needed to cover the testbed.

6.2 A Theoretical Model

In this section we develop a theoretical model that allows us to estimate the radius of a cell such that a HandyMote can still interfere with all nodes in the cell. We will also model the amount of cross-talk between neighboring cells. Finally, we develop a model that allows us to estimate how many HandyMotes are at least needed to cover a testbed deployed over a geographical area A.

In order to derive the models, we need to make a number of practical assumptions. Firstly, we assume that the minimum distance between a pair of motes in the testbed equals D_{min} with typical values in the order of few meters. For example, it is common practice to place a mote in each room on an office floor. Secondly, we assume that we can reduce the RF output power level of the testbed motes to a value P_{mote} below the maximum of 0 dBm (e.g., -10 dBm) without loosing connectivity. In practice, this is often done to obtain multi-hop topologies with a large diameter even on the constrained space of an office floor. Thirdly, we assume that a mote is only able to receive a packet with a certain minimal signal strength P_{min}, with typical values in the order of -90 dBm. Finally, we assume the signal propagation can be modeled with the widely used log-normal model [23, 24, 25]:

$$P(d) = P_T - P_L(d_0) - 10 \cdot \eta \cdot \log_{10} \frac{d}{d_0} + \chi_\sigma \tag{1}$$

where $P_L(d_0)$ is the path loss measured at reference distance d_0, η is the path loss exponent, χ_σ is a zero-mean Gaussian random variable with standard deviation σ that models the random variation of the RSSI value due to shadowing. We use the well-known $P_L(2) = 46$ dBm, and the typical path loss exponent for indoor environments $\eta = 6$ without accounting for shadowing.

Consider the scenario in Figure 12(a) with a HandyMote β and several motes α_i. We are interested in computing the cell radius d_β such that HandyMote β can block the reception of any message by motes contained in its cell (i.e., α_0 and α_1 in the figure). Further, we are interested in the radius Δ_β of the cross-talk region. The cross-talk region is defined as the region where the reception of a message by a mote (i.e., α_2 in the figure) may but need not be blocked by HandyMote β.

Knowing output power P_{handy} of the HandyMote and P_{mote} of the mote, as well as the minimum distance D_{min} between motes, we can compute the maximum RSSI P_{max} a mote can receive from another mote:

$$P_{max} = P_{mote} - P_L(d_0) - 10 \cdot \eta \cdot \log_{10} \frac{D_{min}}{d_0} \tag{2}$$

Using that value and the output power P_{handy} of the HandyMote, we can compute the radius of the cell d_β as follows:

$$d_\beta = 10^{\frac{-P_{max} + P_{handy} - P_L(d_0) + 10 \cdot \eta \cdot \log_{10}(d_0)}{10 \cdot \eta}} \tag{3}$$

Knowing the minimum RSSI P_{min} at which a mote can still receive a message, we can compute the radius of the cross-talk region Δ_β as follows:

$$\Delta_\beta = 10^{\frac{-P_{min} + P_{handy} - P_L(d_0) + 10 \cdot \eta \cdot \log_{10}(d_0)}{10 \cdot \eta}} \tag{4}$$

From that we can compute the difference between the cell radius and the radius of the cross-talk region as $\Theta = d_\beta - \Delta_\beta$.

Knowing the cell radius d_β, we now derive a simple model to estimate the number of HandyMotes needed to cover a given area A. As illustrated in Figure 12(b), we consider the sparsest-possible coverage of an area with disks. Ignoring border effects, the area covered by a single cell can be estimated with the area of the hexagon defined by the intersection points of one circle with the six adjacent circles. Dividing A by the area of the hexagon, we can estimate the number of HandyMotes N needed to cover area A:

$$\mathbf{N} = \frac{A}{\frac{3*\sqrt{3}}{2} * d_\beta^2} \tag{5}$$

We now illustrate those model with concrete examples. If we have a sparse testbed with a distance between nodes of $D_{min} = 4$ meters and transmission powers $P_{mote} = -15$ dBm, $P_{handy} = 0$ dBm, we derive $P_{max} \approx -80$ dBm and the radius of our cells $d_\beta = 8$ meters. This configuration would imply that the size of the cross-talk area $\Theta \approx 4$ meters when using $P_{min} = -90$ dBm.

This cell size is obviously very large, and the consequence would be that in theory only $N = 6$ HandyMotes would be needed to cover a testbed area $A = 750m^2$.

However, with this configuration the cross-talk area Θ is quite large. The accuracy of the regenerated interference may therefore be low as all nodes contained in cross-talk areas are potentially interfered by multiple HandyMotes in neighboring cells with different interference traces. To gain more accuracy, we need to decrease the size of the cross-talk area Θ. This can be achieved by reducing the radius of the cells by means of reducing P_{handy}, which requires more cells and HandyMotes to cover the testbed area. To obtain $\Theta \approx 2$ meters, using the same parameters as above, one would need to use a cell radius of $d_\beta \approx 4$ meters, which would imply that to cover the same testbed area $A = 750m^2$, we would need at least $N = 19$ HandyMotes.

6.3 Practical Testbed Configuration

In this section we describe in a practical procedure how to configure the testbed, i.e., how to select the HandyMotes and how to set their power levels such that mote coverage is maximized and cross-talk between cells is minimized.

1. In a first step, we empirically obtain D_{min}, P_{min}, and P_{max} from the testbed as introduced in the previous section. D_{min} can be obtained directly from the layout of the testbed. P_{min} and P_{max} are measured by having the motes in the testbed sequentially broadcast a message and all others nodes record the maximum and minimum RSSI value.
2. Knowing these parameters, we can compute the maximum cell radius according to Equation 3. We overlay a hexagonal grid as depicted in Fig. 12(b) with cells of the computed radius over the testbed layout, place HandyMotes at the center of the overlay cells (or select testbed motes close to the center of the cells to become HandyMotes), and allocate motes to the HandyMotes based on the cell overlay.
3. Next, we sequentially trigger the selected HandyMotes to generate an interference signal at maximum output power

and check if every mote in the cell of a HandyMote is covered. For this, the motes measure the RSSI noise floor and check if it is larger than $(P_{max} + 3)$ dB.

4. If there are any uncovered motes, we select additional HandyMotes in the vicinity of those motes and return to step 3.
5. In order to reduce cell cross-talk, we reduce the output power levels of the HandyMotes to the minimum value that still guarantees coverage using the same approach as in step 3. If the selected power level is not the maximum power levels, then the power levels higher than the selected one can be used to realized different levels of interference strength. Otherwise, only binary interference can be generated.
6. Finally, one may estimate the quality of the generated configuration by counting the number of motes contained in cross-talk region as follows. The HandyMotes sequentially generate an interference signal at the selected output power. All motes outside of the cell of that HandyMote measure RSSI. If the measured value is larger than P_{min}+ 3 dB, then the mote is contained in a cross-talk region. If the number of motes in cross-talk regions is too high, one may start over with a different initial selection of cells.

This procedure is supported by a program running on the testbed motes during the setup phase. After the configuration is completed, the motes may be programmed with the test application. As part of future work, we plan to further automate this procedure.

7. EVALUATION

In this section, we first evaluate the accuracy with which a HandyMote can regenerate a previously recorded interference trace in the time domain. We then augment an existing sensornet testbed infrastructure with JamLab, and evaluate the accuracy with which the augmented testbed can regenerate a previously recorded interference trace in the spatial domain. Finally, we use JamLab to characterize the performance of MAC protocols under interference.

7.1 Temporal Accuracy

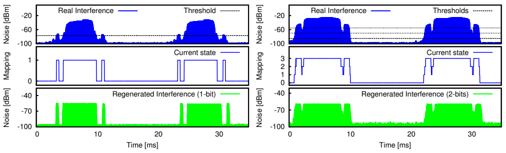
We evaluate the accuracy with which a HandyMote can regenerate a previously recorded interference in the time domain. We run a HandyMote in regeneration mode in proximity of an active Lunik 200 microwave oven warming a bowl of tea. The HandyMote is placed at 1 meter distance from the oven, and records a trace of channel 24 at a sampling rate of 60 kHz. Figure 13(a) (top) shows the interference generated by the microwave as measured by the HandyMote. Next, the trace is quantized to single-bit resolution (middle). Finally, once the microwave oven stopped operating, the HandyMote plays back the recorded binary interference (bottom) using transmission power 0 dBm. As we can notice from the figure, the regeneration is quite accurate in the time domain.

We quantify the accuracy of the regenerated signal with respect to the originally recorded signal using the the *cross-correlation* coefficient (**c**). We represent original and regenerated signals by the series $x(i)$ and $y(i)$, respectively, where $i = 1, \ldots, N$. These series are binary, and take 0 (clear channel) or 1 (busy channel) values. Considering this representation, **c** is given by:

$$\mathbf{c} = \frac{\sum_{i=-\infty}^{\infty} x(i)y(k-i)}{rms(x)rms(y)} \tag{6}$$

where $rms()$ denotes the root mean square value of a signal. We tested eight pairs of original and regenerated samples and the maximum value of **c** was selected for each pair:

$$\mathbf{c}_{xy} = \max_{k \in [-(N-1),(N-1)]} \{\mathbf{c}\} \tag{7}$$



(a) 1-bit Mapping (b) 2-bit Mapping

Figure 13: Regenerated interference of a microwave oven.

The average correlation $\mathbf{c}_{xy}$ is 0.93 with a standard deviation of 0.065. Hence, our implementation does a commendable job with respect to the cancellation of the jitter between sampled and regenerated interference and hence regenerates interference with a fairly high accuracy.

We carry out the same experiment using 2-bit quantization with thresholds -55, -70, and -80 dBm, and we then regenerate the interference using transmission power register levels 31, 7, and 3 (i.e., 0, -10, -25 dBm), respectively. The results match the above ones with binary interference. Figure 13(b) shows the regeneration process when using a two-bit quantization.

7.2 Impact on Packet Reception Rate

In this section we experimentally study the impact of interference on Packet Reception Rate (PRR), comparing the PRR for original, emulated, and regenerated interference signals. We use the same Lunik 200 microwave oven as in the previous experiment, and collect data at the receiver side of a pair of sensor nodes at about 1 meter distance, with the sender transmitting packets at a rate of 128 packets/sec. The sensor just transmits the packet without any clear channel assessments or duty cycling. We place an HandyMote between the two nodes and we run it both in emulation and regeneration mode, once the microwave oven stopped being active.

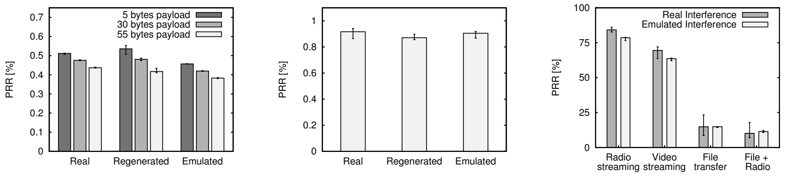
We carry out different experiments with different payload sizes, and we run the HandyMote using transmission power 0 dBm in both emulation and regeneration mode, such that the generated interference signal blocks communication between the sensor nodes.

Figure 14(a) shows the results. The PRR collected when the microwave oven is active decreases when the payload size increases as the probability of periodic microwave interference hitting a packet increases with increasing payload size. The PRR obtained for regenerated interference differs by 5.6% from the original one, hence showing a reasonable accuracy. For emulated interference, the PRR differs from the original one by 12.83%, the reason for that being the noisy amplitude of the original interference signal as depicted in Figure 9, such that occasionally the interference is too weak to block the transmission. In contrast, the emulated interference signal is binary and always blocks communication. Accuracy could be improved in this case by randomly varying the transmission power of the HandyMote as discussed in Section 5.3.

We repeat the experiments in presence of Bluetooth interference. We first measure PRR during a Bluetooth file transfer between a laptop and a mobile phone. We place the HandyMote between the 2 communicating motes and we measure the PRR obtained with original, emulated, and regenerated interference. We run the HandyMote in emulation mode using the models derived in Section 5.2.

Figure 14(b) shows that the packet reception rate obtained under regenerated interference differs by 5.02% from the the original one, while in emulation mode it differs by only 1.31%.

Finally, we repeat the experiment with WiFi interference. Using the same setup as above, we run the HandyMote in emulation mode using the models derived in Section 5.1 while generating WiFi traffic from a laptop according to the scenarios presented on Table 2.



(a) Microwave Oven (b) Bluetooth (c) WiFi

Figure 14: Impact of real, emulated, and regenerated interference on packet reception rate.

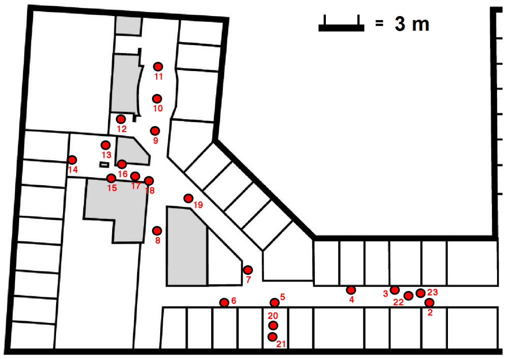


Figure 15: Map of the testbed used for our experiments.

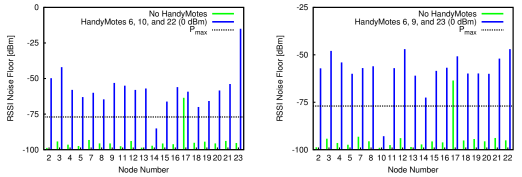
Figure 14(c) shows the results. Also in this case the HandyMote generates interference quite accurately, and the difference between the PRR obtained under real interference and the one obtained under emulation varies between 0.25% and 8.56%. The reason for this difference is that emulation repeats the same pattern over and over, while actual WiFi interference might change in time, due, for example, to TCP adaptation mechanisms.

7.3 Testbed Configuration

Next we want to study the accuracy of the spatial distribution of interference generated by a testbed that has been augmented with JamLab, and therefore configured as described in Section 6.3. Figure 15 shows the topology of the testbed we use, which contains 25 Tmote Sky nodes deployed in an office environment.

As discussed in Section 6.2, there is a tradeoff between the accuracy of regeneration and the cell size, as larger cell size leads to larger cross-talk regions, where motes may be interfered by multiple HandyMotes in neighboring cells. We therefore want to investigate a worst-case scenario with respect to accuracy, where only a few large cells with large cross-talk regions are used. Contrary to the procedure outlined in Section 6.3 to compute the cell size, we therefore start with a cell radius of $d_\beta = 8$ meters, which equals the values in the first example in Section 6.3, where $P_{max} = -80$ dBm. With this cell size, we can cover the testbed with just three cells. We select nodes 6, 10, and 22 as HandyMotes. Next we sequentially trigger the selected HandyMotes to generate interference at maximum output power, and check that the RSSI at every mote is at least $P_{max} + 3dB = -77$ dBm.

Figure 16(a) shows that, with this configuration, node 14 would not be covered as RSSI is smaller than -77 dBm due to the remote location of the node. We therefore change the selection of the HandyMotes (instead of adding more cells) to motes 6, 9, and 23 as shown in Figure 1). With this configuration node 14 is covered, but node 10 is not covered by HandyMote 9 (Figure 16(b)), while in the original configuration node 10 covered node 9 (Figure 16(a)).



(a) HandyMotes 6, 10, and 22 (b) HandyMotes 6, 9, and 23

Figure 16: Testbed augmented with JamLab. In the first configuration, nodes 6, 10, and 22 are selected as HandyMotes. In the second configuration, nodes 6, 9, and 23 are selected instead.

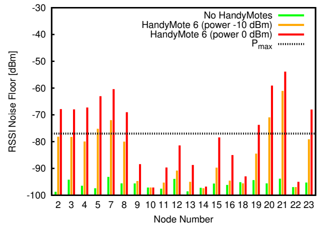


Figure 17: Impact of TX power of HandyMote 6.

This is an example of an asymmetric link – something our simple model in Section 6.2 does not capture. Figure 16 shows also another practical problem. Node 17 is apparently broken as it always returns RSSI readings higher than −67 dBm. We therefore ignore this node in the remainder of the experiments.

Finally, we need to reduce the output power of the HandyMotes to minimize the cross-talk area while still maintaining coverage. The cell controlled by HandyMote 6 is quite small and therefore it is possible to reduce its output power. We show the outcome of varying the transmission power of HandyMote 6 in Figure 17: power level 11 (-10 dBm) is the smallest that provides full cell coverage. Similarly, we obtain output power levels 31 and 7 for HandyMotes 9 and 23, respectively. Figure 18 shows that with such configuration, only the HandyMote controlling a cell can generate an interfering signal at the other motes in the cell exceeding P_{max}.

7.4 Spatial Accuracy

Using the testbed configuration obtained in the previous section, we now study how accurately we can regenerate the spatial distribution of interference. For this, we place a Whirlpool M440 microwave oven in the position marked as M in Figure 1, within the cell controlled by HandyMote 6. This case represents a worst-case scenario, as the oven can interfere over long distances due to its high (60 dBm) and highly varying output power.

Our goal is to record the spatial distribution of the interference patterns generated by the microwave oven in one of the most affected channels (23) all over the testbed. We then let the HandyMotes regenerate the recorded traces while the remaining nodes

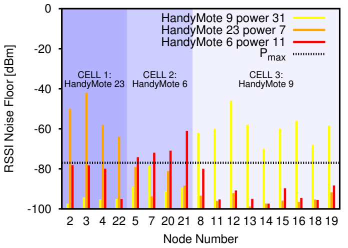
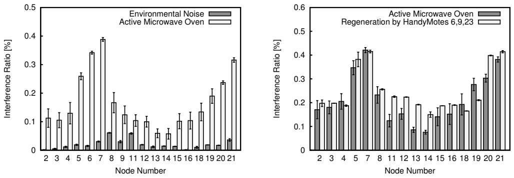


Figure 18: JamLab configuration with independent cells.



(a) Impact of microwave oven (b) Impact of regeneration

Figure 19: Comparison between the interference generated by an active microwave oven and the one regenerated by JamLab in regeneration mode throughout the whole testbed.

record the regenerated interference and compare it with the original interference recorded while the microwave oven was active.

As we have already investigated the temporal accuracy of regeneration in Section 7.1, we now focus on the distribution of the intensity of interference. Instead of recording raw traces, every mote computes the *interference ratio* as the percentage of time in which interference is present (i.e., the percentage of RSSI noise floor readings higher than P_{max}). Figures 19(a) and 19(b) show the comparison of the interference ratio during the activity of the microwave oven, and during the regeneration using JamLab (HandyMotes 6, 9, and 23). Due to their different distances from the microwave oven, node 7 recorded the highest interference ratio when the oven was active, followed by nodes 6, 21, 20, and 5, respectively (Figure 19(a)). The regeneration within this cell is based on the trace recorded by HandyMote 6, therefore nodes 21, 20, and 5 will perceive a higher interference ratio, node 7 a lower one (Figure 19(b)). A similar reasoning can be applied to all other nodes in the testbed: node 14, for example, perceives a higher interference ratio, as recorded by HandyMote 9, which is closer to the oven. If a better spatial accuracy is required, a higher number of (smaller) cells needs to be configured, as discussed in Section 6.

It is important to remark that the environmental noise may play an important role in the quality of the (re)generation, as it will add-up to the interference (re)generated by the HandyMotes. Observing Figures 19(a) and 19(b), we can see how the interference received by node 8 is higher than the one recorded by HandyMote 9 due to a high environmental noise. In order to reduce the non-determinism caused by differences in ambient interference between recording and regeneration, the experiments should be run when the background noise is low, for example in the evening or during the night.

We finally investigate the accuracy of the regeneration with respect to PRR. We repeat the above experiment while nodes pairs (2,3), (5,21), and (18,19) transmit and receive packets with a payload of 5 bytes at a rate of 64 packets/second on channel 23. Figure 20 shows that PRR values are similar between original and regenerated interference for the first two pairs of nodes, while there is a larger error (31.6%) for pair (18,19). This is due to nodes 18 and 19 being much closer to the microwave oven than HandyMote 9, following the discussion made for Figure 19.

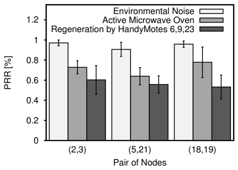


Figure 20: Comparison of the PRR obtained generating interference using a microwave oven and using JamLab.

7.5 Characterization of Protocol Performance

In this section we demonstrate the usability of JamLab by characterising the impact of interference on low-power MAC protocols. We show that using JamLab we can get important insights regarding protocol behaviour under *emulated but realistic* interference.

We perform our experiments in the testbed shown in Figure 15. Our setup consists of a sender (node 7), a receiver (node 5) and one (node 6) or two (nodes 6 and 21) HandyMotes, whose position and transmission power is carefully chosen to jam the communications between sender and receiver. The sender transmits 400 packets to the receiver at a rate of 1 packet/sec. We use 3 different MAC layers: NULLMAC, a simple layer that just forwards packets between the radio driver and the network layer, X-MAC [26], and X-MACQ [27], an enhanced X-MAC with a queue and the ability to rapidly drain the queue in absence of interference. A first HandyMote generates interference using the implementation of Garetto's 802.11 model presented in Section 5.1. We use the model with the RTS/CTS access mechanism and set the minimum and maximum contention window size to 32 and 1032, respectively, as these seem to be the most widely used parameter settings. We emulate saturated traffic from 20 stations (this amount was chosen to have an interference time similar to the one of an active microwave oven), where each station sends packets with a size of 1000 Bytes. A second HandyMote emulates a microwave oven, as in Section 5.1.

Table 3 shows our results. We depict the average results of three runs. With NULLMAC and microwave oven interference, the PRR is slightly lower than 50%, which confirms the results in Figure 14(a). The table also shows that under microwave oven interference, X-MAC performs better than NULLMAC with smaller payloads. As explained in [27], the reason for this is X-MAC sending strobes for a longer time than its off-time and hence the receiver has on average more than one chance to complete the handshake.

While it is known that the PRR decreases with increasing packet size, X-MAC's PRR decreases significantly, namely from almost 60% to less than 40%, as the packet size increases from 30 to 100 Bytes. Also in presence of WiFi interference X-MAC performs much worse for large packet sizes. The experimental results in Figure 14(a) are taken using NULLMAC. Combining the results in this figure with the ones in Table 3, we see a very modest decrease of NULLMAC's PRR with increasing packet size.

The difference between NULLMAC and X-MAC is that in order to receive a data packet, a receiver that employs NULLMAC needs to successfully receive 1 packet only, whereas X-MAC requires the completion of the handshake, i.e., the receiver needs to receive the sender's strobe and acknowledge it, before the sender can send the data packet to the receiver. In our experiments, this data exchange must happen within one time period without interference. This means that until the data packet itself is transmitted, a substantial fraction of a time period without interference has already been used for the handshake. Note that this time period without interference is short due to the bursty interference patterns created by both microwaves and Garetto's WiFi model as the latter emulates saturated traffic. This explains why the packet size is

Payload (Bytes)	Oven NULL	Oven X-MAC	WiFi X-MAC	Both X-MAC	Both X-MACQ
30	45.3%	59.2%	41.6%	20.9%	39.7%
100	43.6%	39.5%	23.8%	9.2%	15.6%

Table 3: Performance of different MAC protocols under emulated but realistic interference (average PRR in %)

more important for X-MAC than for NULLMAC. The table also shows that X-MACQ is more robust than X-MAC against interference, hence confirming the results in [27] using more realistic interference patterns generated using JamLab.

8. RELATED WORK

The study of interference sources in the ISM band has received significant attention from the research community, especially in the crowded 2.4 GHz band. Petrova et al. [1] perform measurements using 802.11g/n devices and quantify their impact on 802.15.4 networks. Sikora reports the impact of microwave ovens, 802.11, and Bluetooth on the packet reception rate of 802.15.4 networks [2]. Liang et al. present a careful analysis of the symmetric and asymmetric IEEE 802.15.4 and 802.11 interference patterns [28]. The high number of interference sources in the ISM band has motivated the study of solutions to overcome interference, in particular WiFi [17, 29]. While we evaluate the same sources of interference, the distinctive and most important contribution of our work is that we provide a low-cost tool to (re)create interference in sensornet testbeds, which goes beyond a one-time-evaluation approach, and enables a better study and debugging of communication protocols.

To the best of our knowledge, we are the first to develop such a low-cost testbed framework for the generation of controlled and realistic interference. Existing sensornet testbeds do not provide any capability for interference generation, or they are limited to static WiFi access points randomly placed in the testbed [3]. JamLab, instead, can seamlessly augment existing sensornet testbeds to study the robustness of protocols against interference.

We have recently discussed the idea of using the CC2420 test modes to generate interference [14, 15] in conjunction with a directional antenna to direct the interference towards a selected set of motes. Our present work goes significantly beyond this, by providing accurate RSSI readings, record-and-playback and emulation of interference capabilities, as well as the integration of these functions into a testbed for interference studies.

Several studies have evaluated the impact of interference on the performance of MAC protocols [27, 30], and a set of fair transmission schedules have been derived by synchronizing the transmission of neighboring nodes in the presence of interference [31]. This type of studies would definitely benefit from the realistic interference patterns that JamLab provides.

9. CONCLUSIONS AND FUTURE WORK

Interference has a strong impact on the performance of sensor networks. Hence, protocols need to be tested under realistic and controlled interference. We present JamLab, a tool to augment existing sensornet testbeds with a low-cost infrastructure for the creation of realistic and repeatable interference patterns. JamLab provides simple models to emulate the interference patterns generated by several devices, and a playback capability to regenerate recorded interference patterns. We demonstrate the utility of JamLab by showing its accuracy in both temporal and spatial domains.

Future work includes a further automation of the testbed configuration procedure, and an accurate study and modeling of new interference sources in the frequency bands used by sensornets.

Acknowledgments

We thank our Shepherd Dr. Yuvraj Agarwal for his constructive comments. This work has been supported by the European Commission with contracts FP7-ICT-224282 (GINSENG) and FP7-2007-2-224053 (CONET), by VINNOVA, by SSF and by the DFG-funded Cluster of Excellence 306/1 "Inflammation at Interfaces".

10. REFERENCES

[1] M. Petrova et al. Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks. In *International Conference on Networking 2007*.

[2] A. Sikora and V.F. Groza. Coexistence of IEEE 802.15.4 with other systems in the 2.4 GHz-ISM-Band. In *IEEE Instrumentation and Measurement Technology*, May 2005.

[3] Divya Sakamuri. NetEye: a Wireless Sensor Network Testbed. Master's thesis, Wayne State University, 2008.

[4] Texas Instruments. *Smart RF CC2420 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver*, March 2007.

[5] A. Dunkels and B. Grönvall and T. Voigt. Contiki - a Lightweight and Flexible OS for Tiny Networked Sensors. In *EmNetS'04*.

[6] The Contiki Projects Community. http://sourceforge.net/projects/contikiprojects.

[7] Y. Liu et al. IEEE 802.11 WLANs WG Information doc. no. 802.11-10/1079r0, September 2010.

[8] P. Verkaik, Y. Agarwal, R. Gupta, and A. Snoeren. SoftSpeak: Making VoIP Play Fair in Existing 802.11 Deployments. In *NSDI'09*.

[9] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys'04*.

[10] Anritsu MS2711D Spectr. Analyzer. http://www.anritsu.com.

[11] N. Tsiftes, A. Dunkels, Z. He, and T. Voigt. Enabling Large-Scale Storage in Sensor Networks with the Coffee File System. In *IPSN'09*.

[12] Q. Wang and T. Zhang. Source Traffic Modeling in WSN for Target Tracking. In *Proc. of the 5th ACM PE-WASUN*, 2008.

[13] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *Proc. of EmNets'07*.

[14] C.A. Boano et al. Controllable Radio Interference for Experimental and Testing Purposes in WSN. In *IEEE SenseApp 2009*.

[15] C.A. Boano, K. Römer, Z. He, T. Voigt, M. Zuniga, and A. Willig. Generation of Controllable Radio Interference for Protocol Testing in Wireless Sensor Networks. In *SenSys'09, demo session*.

[16] Flaviu Cristian. Probabilistic clock synchronization. *Distributed Computing*, 3(3):146 – 158, 1989.

[17] R. Musaloiu-E. and A. Terzis. Minimising the effect of WiFi interference in 802.15.4 WSN. *IJSNet'07*, 3(1):43–54.

[18] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.

[19] M. Garetto and C.F. Chiasserini. Performance analysis of 802.11 WLANs under sporadic traffic. In *Networking'05*.

[20] P. Rathod et al. Characterizing the exit process of a non-saturated IEEE 802.11 wireless network. In *MobiHoc'09*.

[21] M. Vollmer. Physics of the microwave oven. In *Physics Education 39/1*, pages 74–81. IOP Publishing Ltd, 2004.

[22] T. Taher, M. Misurac, J. LoCicero, and D. Ucci. Microwave oven signal modelling. In *WCNC'08*.

[23] D. Lymberopoulos, Q. Lindsey, and A. Savvides. An Empirical Characterization of Radio Signal Strength Variability in 3-D IEEE 802.15.4 Networks Using Monopole Antennas. In *EWSN'06*.

[24] M. Zuniga and B. Krishnamachari. Analyzing the Transitional Region in Low-Power Wireless Links. In *SECON'04*.

[25] E. Miluzzo, X. Zheng, K. Fodor, and A. Campbell. Radio characterization of 802.15.4 and its impact on the design of mobile sensor networks. In *Wireless Sensor Networks*, volume 4913, 2008.

[26] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled WSN. In *SenSys'06*.

[27] C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M. Zuniga. Making Sensornet MAC Protocols Robust Against Interference. In *EWSN'10*.

[28] C. Liang, N. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *SenSys'10*.

[29] J. Hauer, V. Handziski, and A. Wolisz. Experimental Study of the Impact of Wlan Interference on IEEE 802.15.4 BAN. In *EWSN'09*.

[30] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.M. Liang, and A. Terzis. Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless. In *SenSys'10*.

[31] Y. Yi, G. de Veciana, and S. Shakkottai. On optimal MAC scheduling with physical interference. In *INFOCOM'07*.

Paper D

C.A. Boano, M.A. Zúñiga, K. Römer, and T. Voigt. **JAG: Reliable and Predictable Wireless Agreement under External Radio Interference.** *In Proceedings of the 33rd IEEE International Real-Time Systems Symposium (RTSS).* San Juan, Puerto Rico. December 2012.

Summary. This paper investigates the problem of agreement under external radio interference. First, it points out the limitations of traditional message-based approaches. Second, this paper illustrates the design and implementation of JAG, a novel protocol that uses jamming instead of message transmissions to make sure that two neighbouring nodes agree. Third, this paper shows that JAG outperforms message-based approaches in terms of agreement probability, energy consumption, and time-to-completion; and that JAG can be used to obtain performance guarantees and meet the requirements of applications with real-time constraints.

My contributions. I am the main author of this paper and I conceived the idea that detecting a jamming sequence in the presence of external interference is more reliable than using acknowledgement packets. I designed and implemented JAG entirely using Contiki, and I wrote the vast majority of the paper in collaboration and discussion with the co-authors. All the experiments in the evaluation section were carried out by me, whereas the probabilistic model bounding the fraction of disagreements and positive agreements presented in Section V has been created by Marco Zuniga. I presented the paper at RTSS'12.

©2012 IEEE.
ISBN: 978-1-4673-3098-5.
DOI: 10.1109/RTSS.2012.82.

Link: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6424814.

3. In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

JAG: Reliable and Predictable Wireless Agreement under External Radio Interference

Carlo Alberto Boano[†], Marco Antonio Zúñiga[§], Kay Römer[†], and Thiemo Voigt[¶‡]

[†]Institute of Computer Engineering, University of Lübeck, Lübeck, Germany
[§]Embedded Software Group, Delft University of Technology, Delft, The Netherlands
[¶]Department of Information Technology, Uppsala University, Uppsala, Sweden
[‡]Swedish Institute of Computer Science, Kista, Sweden
E-Mail: [†]{cboano, roemer}@iti.uni-luebeck.de, [§]m.zuniga@tudelft.nl, [‡]thiemo@sics.se

***Abstract*—Wireless low-power transceivers used in sensor networks typically operate in unlicensed frequency bands that are subject to external radio interference caused by devices transmitting at much higher power. Communication protocols should therefore be designed to be robust against such interference. A critical building block of many protocols at all layers is *agreement* on a piece of information among a set of nodes. At the MAC layer, nodes may need to agree on a new time slot or frequency channel; at the application layer nodes may need to agree on handing over a leader role from one node to another. Message loss caused by interference may break agreement in two different ways: none of the nodes uses the new information (time slot, channel, leader) and sticks with the previous assignment, or – even worse – some nodes use the new information and some do not. This may lead to reduced performance or failures.**

In this paper, we investigate the problem of agreement under external radio interference and point out the limitations of traditional message-based approaches. We propose JAG, a novel protocol that uses jamming instead of message transmissions to make sure that two neighbouring nodes agree, and show that it outperforms message-based approaches in terms of agreement probability, energy consumption, and time-to-completion. We further show that JAG can be used to obtain performance guarantees and meet the requirements of applications with real-time constraints.

***Keywords*-Acknowledgement; Agreement; Handshake; JAG; Jamming; Radio Interference; Two Generals' Problem; Wireless Sensor Networks.**

I. Introduction

Wireless sensor nodes often need to agree on fundamental pieces of information that can drastically affect the performance of the entire network. For example, sensor nodes may need to agree on handing over a leader role from one node to another. An agreement failure would break the leader election, leading to a situation in which either more than one node becomes leader, or no leader is selected, causing reduced performance or failures in the network [1]. Similarly, at the MAC layer, several state-of-the-art protocols use time division multiple access (TDMA) or frequency diversity techniques to optimize their performance, in order to maximize network lifetime and minimize battery depletion. In such protocols, vital information such as the TDMA schedule, the channel-hopping sequence derived by interference-aware protocols, or the seed used to regulate the random channel hopping, need to be agreed upon by two or more sensor nodes in a reliable fashion. Failure to agree on such information correctly (e.g., nodes using inconsistent TDMA schedules) may disrupt network connectivity or substantially degrade performance.

When sharing information using an unreliable medium (such as wireless), no delivery guarantee can be given on the messages that are sent. Akkoyunlu et al. [2] have shown that, in an arbitrary distributed facility, it is impossible to provide the so called *complete status*, i.e., one cannot guarantee that two distributed parties know the ultimate fate of a transaction and whether they are in agreement with each other.

The problem is further exacerbated in the presence of external interference: the low-power transmissions of wireless sensor networks are highly vulnerable to interference caused by radio signals generated by devices operating in the same frequency range. Several studies have highlighted the increasing congestion of the unregulated ISM bands used by wireless sensor networks to communicate, especially the 2.4 GHz band [3]. Sensornets operating on such frequencies must cope with simultaneous communications of WLAN and Bluetooth devices, as well as with the electromagnetic noise generated by domestic appliances such as microwave ovens, video-capture devices, or baby monitors. As a result, wireless sensor nodes often communicate through interfered channels that have low chances of successfully delivering a packet. Hence, it is important to derive reliable techniques to ensure agreement even in the presence of interference, and make sure that they are efficient enough to meet the limited computational capabilities and energy resources of sensor nodes.

In this work, we design, implement, and evaluate JAG, a simple yet efficient agreement protocol for wireless sensor networks exposed to external interference. JAG introduces a jamming sequence as the last step of a packet handshake between two nodes to inform about the correct reception of a message carrying the information to be agreed upon. The key insight behind this approach is that detecting a jamming sequence in the presence of external interference is more reliable than using acknowledgement (ACK) packets to verify whether the information was successfully shared.

In environments that experience high levels of external interference, the probability of successfully transmitting a sequence of packets and completing an handshake is small, even when using short ACK packets. Despite the minimal amount of information they carry, acknowledgements are embedded into IEEE 802.15.4 frames, and hence can be destroyed if any of the bits in the header, payload, or footer is corrupted by interference. Performance can be improved by means of redundancy (i.e., by sending multiple ACK packets), but this results in a significantly higher energy expenditure and latency, which is undesirable when using resource-constrained wireless sensor nodes.

Using JAG, instead, one can minimize the energy expenditure and provide agreement guarantees under weaker and more realistic assumptions about the underlying interference pattern compared to message-based approaches. By appropriately tuning the length of the jamming sequence, one can parametrize JAG to obtain predictable performance and to guarantee agreement in a finite amount of time, even in the presence of external interference: a perfect fit for applications with timeliness requirements. We focus on the unicast case (agreement between two neighbouring nodes) and show that JAG outperforms traditional packet-based agreement protocols in the presence of interference with respect to agreement probability, energy consumption, and time-to-completion.

JAG is intended as a building block to construct protocols at different layers of the protocol stack. It could be embedded into a MAC protocol to agree on time slots or frequency channels as discussed in Sect. VII, at the transport level to agree on connection establishment or tear-down, or at the application level to agree on handover of a leader role.

Our paper proceeds as follows. Sect. II defines the agreement problem in wireless sensor networks challenged by external radio interference. Sect. III conveys the main idea of the paper: using jamming as a binary signal for acknowledging the reception of packets. Thereafter, in Sect. IV, we illustrate JAG, a protocol for reliable agreement under external radio interference. We describe how JAG can provide the desired quality of service (QoS) in Sect. V, and we experimentally evaluate the performance of JAG under interference in Sect. VI. After discussing the integration of JAG into existing sensornet MAC protocols in Sect. VII, we review related work in Sect. VIII and conclude our paper in Sect. IX.

II. Problem Definition

Agreeing on a given piece of information is a classical coordination problem in distributed computing. The *Two Generals' Agreement Problem*, formulated by Jim Gray to illustrate the two-phase commit protocol in distributed database systems [4], is often used to explain the challenges when attempting to coordinate an action by communicating over a faulty channel, and can be described as follows.

Two battalions are encamped near a city, ready to launch the final attack. Because of the redoubtable fortifications, the attack must be carried out by both battallions at the same time in order to succeed. Hence, the generals of the two armies need to agree on the time of the attack, and their only way to communicate is to send messengers through the valley. The latter is occupied by the city's defenders, and a messenger can be captured and its message lost, i.e., the communication channel is unreliable. Since each general must be aware that the other general has agreed on the attack plan, messengers are used also to exchange acknowledgements. However, because the acknowledgement of a message receipt can be lost as easily as the original message, a potentially infinite series of messages is required to reach an agreement[1].

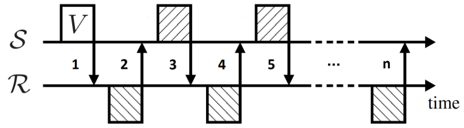


Fig. 1. n-way handshake between nodes $\mathcal{S}$ and $\mathcal{R}$.

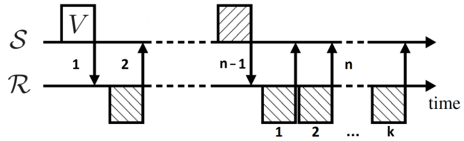


Fig. 2. Enhanced n-way handshake between nodes $\mathcal{S}$ and $\mathcal{R}$ using redundancy: the last ACK message is transmitted k times.

A. Agreement in Wireless Networks

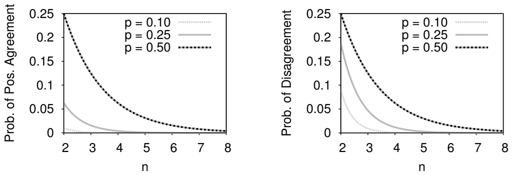
In the context of wireless communications, the problem can be rephrased as follows. When two nodes, $\mathcal{S}$ and $\mathcal{R}$, need to agree on a common value V, they exchange a sequence of n messages in an alternating manner (Fig. 1). Node $\mathcal{S}$ is the initiator of the exchange. After the transmission of V, each subsequent message acknowledges the receipt of the previous message, i.e., a node sends message $i > 1$ only if it correctly received message $i-1$. Each node uses a simple rule to determine the success of the exchange: if all expected messages are received, the exchange is deemed successful, otherwise the exchange is deemed unsuccessful.

The scenario described above corresponds to an *n-way handshake* between nodes $\mathcal{S}$ and $\mathcal{R}$, where n is the number of packets exchanged. The n-way handshake is a widely used mechanism in communication networks. For example, TCP employs a 3-way handshake ($n=3$) to establish connections over the network, whereas IEEE 802.11i (WPA2) uses a 4-way handshake ($n=4$) to carry out the key exchange.

An n-way handshake can have three possible outcomes:

1) **Positive Agreement.** The n messages are all received correctly, and both nodes deem the exchange as successful, accepting V.
2) **Negative Agreement.** A message m with $m < n$, i.e., a message prior to the last message n, is lost. None of the nodes receives all the expected messages, hence both nodes deem the exchange as unsuccessful, discarding V.
3) **Disagreement.** The last message n is lost. One of the two nodes receives all the expected messages, deems the exchange as successful and accepts V; whereas the second node misses the last message and therefore deems the exchange as unsuccessful, rejecting V.

[1] A different problem that we are not addressing in this work is how to guarantee the identity of the sender of the message, as well as how to cope with misbehaving parties.



(a) Positive Agreement

(b) Disagreement

Fig. 3. Distribution of the probabilities of positive agreement and disagreement of the n-way handshake shown in Fig. 1 as a function of the probability of successful packet transmission p and length of the handshake n.

In the original two generals' scenario, a *positive agreement* would lead to a simultaneous attack of the city by both battalions and a consequent victory, a *negative agreement* would cause both battalions to stall, while a *disagreement* would trigger the attack of only one battalion and a consequent defeat of the attacking forces.

While *disagreements* are potentially fatal, *negative agreements* are often less severe. For example, if the shared value contains the next channel to be used for communication, two nodes are better off staying in the same lossy channel, rather than having only one of them move to a different frequency. The probability of negative agreements should, however, be minimized, as it may lead to reduced performance. Hence, an agreement protocol should strive to minimize disagreement as a first priority, maximize positive agreements as a second (almost equally high) priority, and minimize negative agreements as a third (substantially lower) priority. A metric to measure the quality of an agreement protocol (whose value should be minimized) is therefore the *DPA ratio* of the probability of disagreements over the probability of positive agreements.

B. The importance of the last message

It is important to emphasize that, in an n-way handshake, disagreements only occur if the last message is lost. Hence, depending on the application, it may be desirable to devote extra-resources to increase the successful delivery of the last packet by means of redundant packet transmissions (i.e., repeating a message several times and assuming successful transmission if at least one copy is received).

A possibility is to employ a n-way handshake in which the last packet is repeated k times, as shown in Fig. 2. Using this approach, the final outcome of the handshake is strongly dependent on the link quality, on the length n of the n-way handshake, and on the redundancy factor k. Letting p represent the probability that a generic message is successfully received (assuming that p remains constant over time and that it is independent for each packet), and $q = 1 - (1-p)^k$ the probability of successfully receiving at least one of the k redundant packets, we obtain:

$$\begin{aligned}\text{Prob}(PositiveAgreement) &= p^{n-1}q \\ \text{Prob}(NegativeAgreement) &= 1 - p^{n-1} \\ \text{Prob}(Disagreement) &= p^{n-1}(1-q)\end{aligned}$$

These equations show that in order to maximize the frequency of positive agreements and, at the same time, minimize the frequency of disagreements, we need to maximize the link quality p and maximize the level of redundancy k. The choice of a suitable n becomes a catch-22 dilemma in the presence of unreliable links, as illustrated in Fig. 3: long n-way handshakes minimize the probability of disagreement, but also the probability of positive agreement, whereas short n-way handshakes maximize the probability of positive agreement, but also the chances of disagreement.

C. Agreement in Wireless Sensor Networks Challenged by External Interference

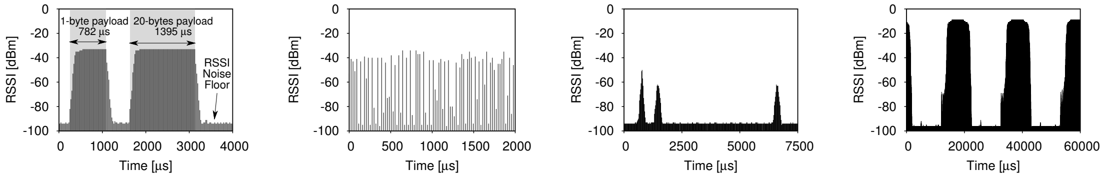
In the context of wireless sensor networks, minimizing the amount of exchanged packets is mandatory because of the limited energy resources available, i.e., sensor nodes need to minimize the time during which the radio is active as much as possible. Therefore, the use of redundant packet transmissions and long handshakes is not advisable, as it would increase the energy consumption.

Another aspect is the channel quality affecting p. Wireless sensor nodes operate in the unlicensed ISM radio bands, and often use a very low transmission power, which makes them vulnerable to external interference. Any wireless appliance operating in the same frequency range of sensornets can potentially interfere with their communications and decrease the probability of a successful packet exchange p. In the 2.4 GHz ISM band, for example, Wi-Fi and Bluetooth networks, as well as domestic appliances such as microwave ovens, can create noise levels that overwhelm the interference resistance capabilities of DSSS radios and radically decrease the packet reception rate [3], [5]. Hence, we need to investigate ways to encode transmissions such that their success probability p is maximized despite interfered channels.

D. Analysis of Common Interference Sources

In order to understand the impact of external interference on the probability of successful transmission p in wireless sensor networks communications, we study the interference patterns produced by common devices operating in the 2.4 GHz ISM band. Using Sentilla Tmote Sky nodes employing a CC2420 radio, we perform a high-speed sampling of the RSSI register ($\approx$ 50 kHz as in [6]). We call this operation *fast RSSI sampling* over a time window t_{samp}. Fig. 4 shows the outcome of fast RSSI sampling in the presence of sensornet communications and external interference.

Absence of external interference. When neither interference nor IEEE 802.15.4 communications are present, the fast RSSI sampling returns the so called RSSI noise floor. The latter has typically values in the proximity of the radio sensitivity threshold (e.g., in the range $[-100, -94]$ dBm for the CC2420 radio). In the presence of IEEE 802.15.4 communications, the fast RSSI sampling returns a stable value corresponding to the strength and the length of the transmitted packet (Fig. 4(a)). As packets have a constrained maximum payload size of 127 bytes according to the 802.15.4 PHY standard, a packet transmission at 250 Kbit/sec would not last more than 4.3 ms.



(a) IEEE 802.15.4 packets (b) Heavy Wi-Fi Interf. (c) Bluetooth interf. (d) Microwave oven interf.

Fig. 4. RSSI values measured using off-the-shelf wireless sensor nodes operating in the 2.4 GHz ISM band. Please notice the different scale of the x-axis.

Presence of external interference. When other devices operating in the same frequency band of wireless sensor networks are active, bursts of interference signals (*busy periods*) alternate with instants in which the channel is clear (*idle periods*). The strength of the interference signals and the duration of idle and busy periods depend on the interfering source and on the specific context. For example, the interference patterns generated by Wi-Fi transmissions depend on the number of active users and their activities, as well as on the traffic conditions in the backbone.

Wi-Fi transmissions are typically much stronger than sensornet transmissions, and can affect several IEEE 802.15.4 channels at the same time. Hauer et al. [7], [8] have shown that with a sufficiently high sampling rate, one can identify the short instants in which the radio medium is idle due to the Inter-Frame Spaces (IFS) between 802.11 b/g packets. Fig. 4(b) shows the outcome of fast RSSI sampling in the presence of heavy Wi-Fi interference (caused by a file transfer): it is indeed possible to identify RSSI values matching the radio sensitivity threshold between consecutive Wi-Fi transmissions.

Fig. 4(c) shows an example of interference generated by Bluetooth. The latter uses an Adaptive Frequency Hopping mechanism to combat interference, and hops among 1-MHz channels around 1600 times/sec., hence it remains in a channel for at most 625 μs. Since Bluetooth channels are more narrow than the ones defined by the 802.15.4 standard, it may happen that communication in multiple adjacent Bluetooth channels affects a single 802.15.4 channel.

Fig. 4(d) shows an example of the interference pattern caused by microwave ovens: high-power noise ($\approx$ 60 dBm) is emitted in the 2.4 GHz frequency band in a very periodic fashion. The period mostly depends on the power grid frequency, but can also slightly vary depending on the oven model. Works in the literature report a power cycle of roughly 20 ms (at 50 Hz) or 16 ms (at 60 Hz) with an active period of at most 50% of the power cycle [6], [9].

E. The Role of Idle Periods

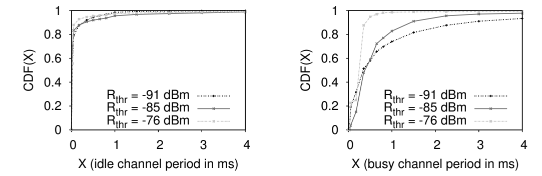
In the presence of external interference, n-way handshakes need to take advantage of idle periods. In principle, the longer the idle period and the shorter the handshake, the higher the likelihood of obtaining positive agreements. However, the interplay between idle periods and n-way handshakes is complex because of the particular patterns of each interfering source. Some devices, such as microwave ovens, generate periodic interference patterns with relatively long idle periods (Fig. 4(d)), while others, such as Wi-Fi stations, generate interference patterns with short idle periods of a highly variable length (Fig. 4(b)).

Having short idle periods reduces the probability of successfully completing a handshake, and this is especially critical in the presence of heavy Wi-Fi interference. Fig. 5 shows the cumulative distribution function (CDF) of idle and busy periods measured by a Maxfor MTM-CM5000MSP node in the presence of a laptop continuously downloading a file from a nearby access point. A channel is defined as busy if the RSSI is higher or equal than a configurable threshold R_{thr} and idle otherwise. In such a scenario, the probability of having an idle period longer than 2 ms is smaller than 5%. Therefore, there is only a little chance that a message-based handshake successfully completes within an idle period. In order to escape interference, one would need to use short messages and send them as close as possible to each other, in order to increase the chances of fitting into an idle period.

Off-the-shelf IEEE 802.15.4-compliant radios such as the CC2420 offer the ability to automatically generate and send ACKs for data frames in hardware. The advantage of hardware acknowledgements is a significant reduction of latency compared to solutions in which the ACK is generated via software [10]. However, hardware ACKs cannot be used to carry out a complete n-way handshake (with $n > 2$), since they cannot be used in reply to another hardware ACK. Imagine a node $\mathcal{S}$ starting a handshake by sending a message to $\mathcal{R}$. The latter can reply with a hardware ACK, but $\mathcal{S}$ will have to receive and extract the packet, analyse its validity, as well as to prepare a new ACK frame, load it into the buffer, and send it over-the-air[2]. This may cause long latencies that break the agreement in the presence of short idle periods.

Furthermore, it is also highly inefficient to encode the binary information carried by an ACK message inside an IEEE 802.15.4 frame, especially in the presence of interference. Despite the payload contains only a single ACK bit, the whole packet consists of synchronization preamble and a physical header (4-bytes preamble, 1-byte Start of Frame Delimiter (SFD), 1-byte length field), as well as a MAC header and footer (2-bytes frame control, 1-byte sequence number, 4-20-bytes address, 2-bytes Frame Check Sequence (FCS)). If any of the bits in the headers and preamble is corrupted by interference, the packet may become undecodable [11], [12].

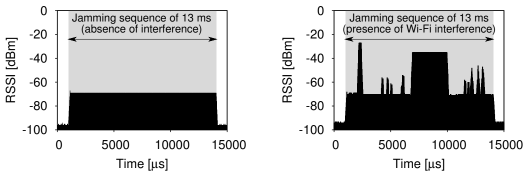
[2] In case a train of k redundant software ACKs is sent, the packet can be loaded into the buffer once and sent repeatedly.



(a) Idle periods

(b) Busy periods

Fig. 5. Cumulative distribution function (CDF) of idle and busy periods measured by a Maxfor MTM-CM5000MSP node in the presence of a laptop continuously downloading a file from a nearby access point.



(a) Absence of Interference

(b) Presence of Wi-Fi Interference

Fig. 6. RSSI values measured by a Maxfor MTM-CM5000MSP node during the transmission of a jamming sequence in absence of interference (a), and in the presence of external Wi-Fi interference (b).

Therefore, instead of encoding the last ACK as packet transmission, we propose to encode it by means of **jamming**, where the presence of a jamming sequence signals the receipt of the previous message. The key advantage of this approach is that jamming, as generated by off-the-shelf wireless sensor nodes, can be reliably detected even under interference.

III. Jamming as Binary ACK Signal

We propose to encode the last acknowledgement of a n-way handshake by means of **jamming** (i.e., transmission of a carrier signal), where the presence of a jamming sequence signals the receipt of the previous message. The key advantage of this approach is that precisely timed jamming signals can be generated using off-the-shelf wireless sensor nodes and can be reliably detected even under heavy interference.

A. Generating a Jamming Sequence

In a recent study, we showed that off-the-shelf radios can be used to generate controllable and repeatable jamming signals in specific IEEE 802.15.4 channels by transmitting a modulated or unmodulated carrier signal that is stable over time [6], [13]. This approach is superior to packet-based jamming, as the generated signal is independent of both packet sizes and inter-packet times. We hence generate precisely timed jamming signals by configuring the MDMCTRL1 register, so that the CC2420 radio outputs a continuous modulated carrier signal. The detection of the latter is based on high-frequency RSSI sampling, as discussed next.

B. Detecting a Jamming Sequence

Common radio chips offer the possibility to read the RSSI in absence of packet transmissions. Several researchers have shown that it is a useful way to assess the noise and the level of interference in the environment [5], [8], [14]. RSSI readings close to the sensitivity threshold of the radio indicate absence of interference, whereas values above this threshold identify a packet transmission, or a busy/congested medium (see Fig. 4).

Hence, we use the fast RSSI sampling mechanism mentioned in Sect. II-D to detect the presence or absence of a jamming signal generated by a sensor node. A jamming sequence generated using the method described in Sect. III-A results in a stable RSSI value above the sensitivity threshold of the radio, as shown in Fig. 6(a). Therefore, one can detect if a jamming signal was transmitted by making sure that no RSSI sample falls down to the sensitivity threshold of the radio.

In the presence of additional external interference, the RSSI register will return the maximum of the jamming signal and the interference signal due to the co-channel rejection properties of the radio [6]. Fig. 6(b) illustrates this for a jamming signal sent in the presence of Wi-Fi interference. As we have shown in Sect. II-D, typical interference sources – in contrast to our jamming signal – do not produce continuous interference for long periods of time, rather they alternate between short idle and busy periods. That is, *if the jamming signal lasts longer than the longest busy period of the interference signal, we are unequivocally able to detect the absence of the jamming signal* by checking if any of the RSSI samples equals the sensitivity threshold of the radio. We exploit this property to design JAG, a protocol for reliable agreement under external interference.

C. Identification of the Interfering Source

While a jamming signal can encode the binary acknowledgement information, it cannot encode the identities of sender and receiver as a regular packet would. When carrying out a handshake, however, these identities are already included in the message V to be acknowledged, and therefore are implicitly known to the two nodes, as long as the communication channel remains allocated exclusively for the whole duration of an exchange. In this way, intra-network interference is avoided, and a jamming sequence acknowledging the reception of V can be identified reliably by means of an RSSI threshold, as we discuss in Sect. IV. Any protocol that embeds JAG as a building block for agreement needs to meet this requirement. At the MAC layer, RTS/CTS can be used to allocate the channel in CSMA protocols, whereas in TDMA protocols the timeslots must be long enough to complete an exchange.

IV. JAG: Reliable Agreement under Interference

We call *JAG (Jamming-based AGreement)* the three-way handshake in which the last ACK is sent in the form of a jamming signal as shown in Fig. 7. The choice of three-way handshakes (as opposed to two-way) is motivated by two facts. First, a three-way handshake increases the reliability of identifying the jamming signal because it provides a reference RSSI value (this will be explained in more detail in Sect. IV-B). Second, three-way handshakes avoid disagreements due to asymmetric links: for instance, if $\mathcal{S}$ has a link with $\mathcal{R}$ but the reverse link is not present, a two-way handshake would always lead to disagreements, since $\mathcal{R}$ is not able to confirm the reception of V.

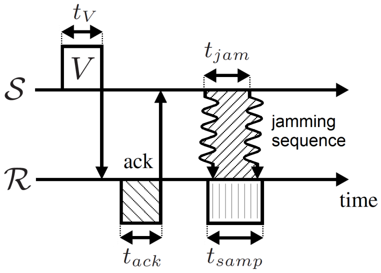


Fig. 7. Illustration of JAG: the last acknowledgement of the 3-way handshake between nodes $\mathcal{S}$ and $\mathcal{R}$ is sent in the form of a jamming signal.

A. *Protocol Design*

The protocol proceeds as follows. $\mathcal{S}$ initiates the exchange and sends the information V towards a receiver $\mathcal{R}$. If V is successfully received, $\mathcal{R}$ saves the signal strength r_s of the received packet and sends an ACK message back to $\mathcal{S}$. We can send either hardware or software acknowledgements: in the remainder of this paper we assume that hardware ACKs are available. If $\mathcal{S}$ receives the acknowledgement, it transmits a jamming signal for a period t_{jam}. Meanwhile, $\mathcal{R}$ carries out a fast RSSI sampling for a period $t_{samp} \leq t_{jam}$ that is synchronized in such a way that the fast RSSI sampling is carried out while the jamming signal is on the air. The message V is used as the synchronization signal: given that clock drift is not too high at timescales of a few milliseconds, it is sufficient to include a short safety margin to compensate for drift (more details in Sect. IV-D). For simplicity, in the rest of the paper, we assume $t_{jam} = t_{samp}$.

If $\mathcal{R}$ detects the presence of the jamming signal, it deems the exchange as successful; otherwise, V is discarded. $\mathcal{S}$ deems the exchange as successful if the ACK is received within a short timeout period, otherwise the jamming sequence is not generated and the handshake immediately terminated.

After the reception of V, node $\mathcal{R}$ carries out a fast RSSI sampling as described in Sect. III to detect the absence or the presence of the jamming sequence transmitted by $\mathcal{S}$. The method to detect the jamming signal is simple: if a jamming sequence is sent, *all* RSSI samples should be above r_{noise}, with the latter being the RSSI noise floor threshold of the radio. Hence, if during t_{samp} we observe *at least one RSSI sample* with a value comparable to r_{noise}, we conclude that the jamming sequence was not transmitted.

This process can be described as follows. Denoting $\{x_1, x_2, \ldots, x_n\}$ as the sequence of RSSI values sampled during t_{samp}, we define the binary sequence $\{X_1, X_2, \ldots, X_n\}$ as follows: if $x_i \leq r_{noise}$, then $X_i = 1$, else $X_i = 0$. $\mathcal{R}$ makes a decision about the presence of the jamming sequence as follows: if $\sum_{i=1}^{n} X_i = 0$, then $\mathcal{S}$ was transmitting a jamming signal and hence V is accepted; otherwise, V is discarded.

Using this algorithm, JAG would operate correctly and would be able to recognize the presence or absence of a jamming signal reliably. However, we can enhance its performance significantly by exploiting the knowledge of the received signal strength r_s of the packet containing V.

B. *The Role of r_s*

Under the hypothesis that the jamming signal has a reasonably similar signal strength to r_s (RSSI does not change significantly between consecutive transmissions spaced by only a few milliseconds), $\mathcal{R}$ can filter out any interference source weaker (i.e., resulting in an RSSI range smaller) than $(r_s - \Delta_r)$, with Δ_r being a tolerance margin to compensate for the inaccuracy of low-power radios and the instability of the RSSI readings. This allows to shorten t_{jam} and achieve a higher energy-efficiency: as we can see in Fig. 5(b), the higher R_{thr}, the shorter the duration of busy periods.

Hence, if $(r_s - \Delta_r) > r_{noise}$, JAG's algorithm is executed as follows: if $x_i < (r_s - \Delta_r)$, then $X_i = 1$, else $X_i = 0$. $\mathcal{R}$ still makes a decision about the presence of the jamming sequence in the following way: if $\sum_{i=1}^{n} X_i = 0$, then $\mathcal{S}$ was jamming and hence V is accepted; otherwise, V is discarded.

Furthermore, r_s also increases the reliability of fast RSSI sampling. The maximum distance over which a packet can be successfully received and decoded is shorter than the distance over which a jamming signal can be captured. This may lead to confusion in a scenario in which two nodes that cannot communicate with each other are allocated the same time slot in a TDMA protocol and transmit a message concurrently. By using a threshold r_s, we make sure that a receiver $\mathcal{R}$ is in the communication range of $\mathcal{S}$, and therefore r_s cannot be achieved by any other node transmitting simultaneously.

C. *The Role of t_{jam}*

The length of the jamming sequence t_{jam} can be tuned in order to provide probabilistic guarantees on the fraction of disagreements. Denoting t_{busy}^{max} as the maximum busy period that can be encountered in the presence of interference, we can guarantee that $\mathcal{S}$ and $\mathcal{R}$ will agree on V by setting $t_{jam} > t_{busy}^{max}$. In such a case, an idle period will surely be encountered during t_{samp}, and the absence of a jamming sequence unequivocally detected, as discussed in Sect. III-B. Hence, the most pernicious outcomes (disagreements) are eliminated, and only positive or negative agreements can occur.

In some scenarios, however, one may need to know the outcome of the agreement process before t_{busy}^{max}. In these cases, where $t_{jam} \leq t_{busy}^{max}$, disagreements may occur. For these type of scenarios, given t_{jam}, we derive an upper bound for the probability of obtaining disagreements. In this way, a user with stringent real-time constraints can assess if the fraction of disagreements is within the limits permitted by the QoS requirements of the application. The probabilistic model bounding the fraction of disagreements is presented in Sect. V.

D. *JAG Implementation*

We implement JAG on Maxfor MTM-CM5000MSP and Sentilla Tmote Sky nodes. Our implementation, based on Contiki [15], uses two main building blocks: the generation of a jamming sequence and the high-frequency RSSI sampling. The former uses the CC2420 transmit test modes as described in Sect. III-A. The latter is implemented as in our previous work [6], so that we roughly obtain one RSSI sample every

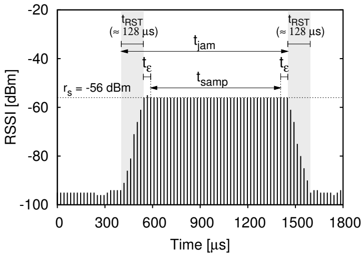


Fig. 8. Alignment between t_{samp} and t_{jam}: RSSI readings obtained during t_{RST} and t_ϵ are discarded to compensate for synchronization inaccuracies.

20 μs. Although a sampling rate of 50 kHz does not capture the transmissions from all wireless devices operating in the same frequency band of sensor networks (e.g., IEEE 802.11n devices), it is still enough to identify most of the idle periods that occur between Wi-Fi transmissions and hence to distinguish the jamming sequence from external interference.

For all our experiments we use NULLMAC, a MAC layer that just forwards packets to the upper or lower protocol layer and does not perform any duty cycling, but reports the presence of hardware acknowledgements. We chose NULLMAC in order to obtain results that are independent of specific MAC features and parameters. To ensure that the execution time of the entire handshake is bounded and independent of clear channel assessment (CCA) back-off times, we do not postpone transmissions until the channel becomes clear. Instead, we carry out a single clear channel assessment before sending V: if the channel is found busy, the transmission is cancelled. This is an optimization, as sending V despite the busy channel would result in a negative agreement (V would be lost).

To ensure alignment between jamming t_{jam} and sampling t_{samp}, we implement a simple synchronization mechanism. $\mathcal{S}$ and $\mathcal{R}$ synchronize their operations based on the reception of V: the transmission or reception of the Start of Frame Delimiter (SFD) is used as the synchronization signal. Although at timescales of a few milliseconds clock drift is minimal, the beginning of t_{samp} may not be aligned with the beginning of the jamming sequence because of the time required for RSSI to settle. The RSSI of the CC2420 radio is indeed an average of the last 8 bit symbols [6] and hence one needs to wait for the RSSI to stabilize (this takes $\approx t_{RST} = 128\mu$s) before being able to measure r_s (see Fig. 8). Since RSSI readings are not instantaneous and their duration may slightly differ among different nodes, we introduce a safety margin t_ϵ during which the RSSI readings are discarded: this allows us to compensate for possible synchronization inaccuracies. The actual length of t_{jam} must therefore be increased by $2 \cdot (t_{RST} + t_\epsilon)$ to make sure that t_{samp} is correctly aligned.

V. Predictable Performance under Interference

We mentioned in Sect. IV-C that one can use t_{jam} to provide probabilistic guarantees on the fraction of disagreements. When setting $t_{jam} > t_{busy}^{max}$ is not possible, it is important to precisely calibrate t_{jam} so that a user with stringent real-time constraints can know in advance the fraction of disagreements to be expected. Hence, we now derive a probabilistic model that bounds the probabilities of positive agreements and disagreements for JAG, given a certain value of t_{jam}.

Variable	Description
t_{pkt}	Transmission delay of PKT containing V
t_{ack}	Transmission delay of ACK
t_{jam}	Duration of jamming signal in JAG
X	Random variable denoting the length of the idle period
$p(x)$	Probability density function (*pdf*) of X

TABLE I
Notation used in our probabilistic model.

The parametrization of the probabilistic model requires the user to run a wireless sniffer in order to capture the characteristics of the surrounding interference. We use continuous RF noise measurements to measure the duration of idle and busy periods and compute their probability density function (*pdf*): a channel is defined as busy if the RSSI is higher or equal than a configurable threshold R_{thr} and idle otherwise.

Preferably, this operation should be carried out before the actual deployment, but it would also be possible to characterize interference at runtime, for example in case the RF environment has changed significantly from the prior observation.

The user can then follow three simple steps: (i) compute the *pdf* of the idle periods $p(i)$, where i represents the length of the idle period, (ii) compute the conditional *pdf* of the busy periods following the idle periods $p(b > x|i)$, and (iii) use the model to obtain the value of t_{jam} that provides the desired QoS.

Table I shows the notation used in our analysis. Our goal is to derive the probabilities of positive agreements and disagreements for JAG given a certain value of t_{jam}. First, we obtain the probability of selecting an idle period of length i, then, we derive the probabilities of obtaining positive agreements and disagreements over all possible idle periods.

Denoting $p(i)$ as the probability density function of the idle periods formed by the interference pattern, the probability of selecting an idle period of length i is given by:

$$s(i) = \frac{ip(i)}{\sum_{i=1}^{\infty} ip(i)} \tag{1}$$

i.e., the more frequent and the longer the idle period, the higher the likelihood of selecting it.

In order to derive the required probabilities, we need to understand the interplay between the length of an idle period i and the 3-way handshake method used by JAG (i.e., the transmission of the PKT embedding V, the ACK, and the JAM signal). In principle, based on the definitions presented in Sect. II, losing an ACK should lead to negative agreements. The practical implementation of JAG, however, takes an optimistic approach that increases the likelihood of positive agreements at the cost of turning some negative agreements into disagreements. In JAG, if $\mathcal{R}$ sends the ACK, four outcomes can occur: (i) a positive agreement, if the ACK is successfully delivered to $\mathcal{S}$ and the JAM signal is correctly decoded by $\mathcal{R}$; (ii) a negative agreement, if the ACK is lost and $\mathcal{R}$ detects the *lack of* JAM; (iii) *another positive agreement*, independently of the fact that the ACK is received or not if, after sending the ACK, $\mathcal{R}$ detects an interference signal with a strength higher than the expected JAM signal and hence assumes a successful transaction (this is the optimistic approach, which assumes the JAM was buried within the stronger signal); and (iv) a disagreement, if the ACK is lost, but, by chance, a high interference signal lasts longer than t_{samp}. In this case, $\mathcal{R}$ assumes, mistakenly, a successful exchange, i.e., a negative agreement turns into a disagreement.

Based on the above description, in JAG, positive agreements are given by the following equation:

$$P_{\text{jam}}\{\text{Pos. Agr.}\} = \sum_{i>t_{\text{pkt}}+t_{\text{ack}}}^{\infty} s(i)\left(1-\frac{t_{\text{pkt}}+t_{\text{ack}}}{i}\right) \tag{2}$$

whereby the first term of the product states the probability of obtaining an idle slot of length i, and the second term states the probability that the selected idle slot can "contain" the transmission of the packet followed by the ACK ($t_{\text{pkt}}+t_{\text{ack}}$).

In order to obtain the fraction of disagreements, we use a bounding probability. There are three necessary but not sufficient conditions to obtain disagreements: (i) PKT is transmitted successfully, (ii) the ACK is corrupted and (iii) the interference signal after the ACK is longer than t_{jam} (to shadow the JAM signal). Hence, we define the probability of obtaining disagreements with JAG as follows:

$$\begin{aligned} P_{\text{jam}}\{\text{Disagreement}\} \leq & \sum_{i=1}^{t_{ack}} s(i)p(b>t_{jam}|i)+ \\ & \sum_{i>t_{ack}}^{t_{pkt}+t_{ack}} s(i)p(b>t_{jam}|i)\left(1-\frac{\min(t_{pkt},i)}{i}\right)+ \\ & \sum_{i>t_{pkt}+t_{ack}}^{\infty} s(i)p(b>t_{jam}|i)\left(\frac{t_{ack}}{i}\right) \end{aligned} \tag{3}$$

Each of the sums on the right side of the equation has three terms. The first term $s(i)$ denotes the probability of obtaining an idle slot of length i. The second term $p(b>t_{jam}|i)$ denotes the probability of obtaining a busy period b longer than t_{jam} after an idle period of length i (the minimum requirement to shadow the jamming signal). The third term differs for each sum, and denotes the probability that the ACK will be corrupted: in the first summation this probability is 1, because the idle time is less than t_{ack}, i.e., the ACK will always be corrupted; in the second and third summations, this probability describes the chances that the agreement starts early enough to allow a successful delivery of PKT, but late enough to corrupt the ACK. Please note that, in Eq. 3, the term $p(b>t_{jam}|i)$ assumes that the corrupted ACK ends exactly before the next busy period starts. In practice, the ACK will likely have a Δ overlap with the beginning of the busy period b, and hence, b will need to be longer than $(t_{jam}+\Delta)$ to lead to a disagreement. Given that $p(b>t_{jam}|i) > p(b>(t_{jam}+\Delta)|i)$, in practice, we can expect a lower fraction of disagreements.

For the case of disagreements, JAG allows the user to fine-tune the duration of t_{jam} according to the requirements of the application (Eq. 3). In Sect. VI-E, we will observe that this fine-tuning capability is central to provide QoS guarantees.

VI. Experimental Evaluation

A. Experimental Setup

We carry out our experiments in two small-scale sensornet testbeds with USB-powered sensor nodes. The first testbed consists of 15 MTM-CM5000MSP nodes deployed in an office environment, whereas the second testbed uses the same type of sensor nodes deployed in a residential building. We use our first testbed to evaluate the performance of several agreement protocols under different types of interference. To this end, we use JamLab [6], a tool for controlled and realistic interference generation in specific IEEE 802.15.4 channels. We configure JamLab to emulate a continuous file transfer produced by either Bluetooth or Wi-Fi devices in specific IEEE 802.15.4 channels. We further carry out experiments in the presence of a Wi-Fi interference generated by a laptop continuously downloading a file from a nearby access point. We validate our first set of results using a second testbed deployed in residential buildings surrounded by Wi-Fi stations: we run different agreement protocols for several days and compare their performance over time.

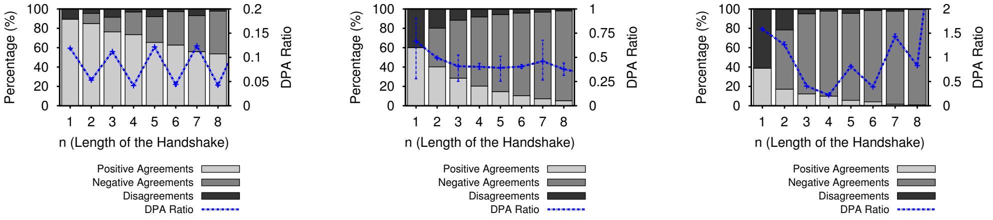
In our experiments, we use several pairs of nodes $\mathcal{S}$ and $\mathcal{R}$. Node $\mathcal{S}$ always initiates the handshake, and transmits a data packet composed of a 6-byte payload containing the information to be agreed upon V and the transmission power used T_P. For each handshake (which is initiated after a random interval in the order of hundreds of milliseconds), we select a random transmission power between -25 dBm and 0 dBm in order to create different types of links. $\mathcal{R}$ replies to the packet using T_P, i.e., the same transmission power used by $\mathcal{S}$. Hardware ACKs are enabled by default, and nodes remain on the same channel during the whole duration of the experiment, in which we perform several hundred thousand handshakes.

B. Packet-based n-way handshake

We firstly analyse the performance of the packet-based n-way handshake shown in Fig. 1 (redundancy factor $k=1$) under different interference patterns. In our implementation, every packet from $\mathcal{R}$ to $\mathcal{S}$ is sent using the hardware ACK support, so to minimize the latency between the reception of the previous packet and the dispatch of the following one.

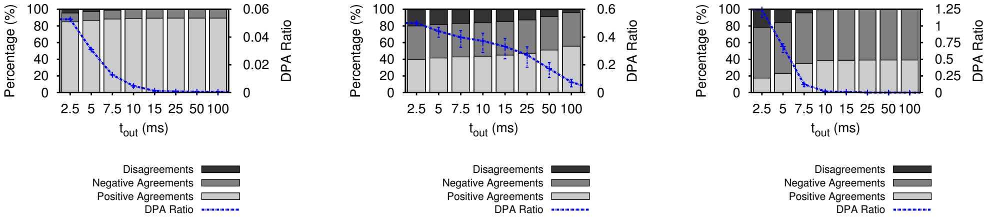
Fig. 9 shows the percentage of positive/negative agreements and disagreements obtained under different interference patterns. The values are computed as an average over all transmission power values T_P used in our experiments, excluding the ones leading to asymmetric links.

Fig. 9(a) depicts the performance of the protocol under JamLab's emulated Bluetooth file transfer. As discussed in Sect. II, the longer the handshake, the smaller the amount of disagreements and positive agreements. Hence, the DPA ratio does not decrease when increasing the length of the handshake n. The alternating performance of the DPA ratio is caused by the interchange between software and hardware ACKs: the former require a higher latency to be transmitted, and hence offer a worse performance with respect to the latter. Fig. 9(b) and 9(c) show the performance of the n-way handshake protocol under JamLab's emulated Wi-Fi transfer



(a) Bluetooth (b) Real Wi-Fi (c) Emulated Wi-Fi

Fig. 9. Performance of a packet-based n-way handshake under different types of interference.



(a) Bluetooth (b) Real Wi-Fi (c) Emulated Wi-Fi

Fig. 10. Performance of 2-MAG (2-way handshake in which the last acknowledgement packet is sent k times) under different types of interference. The longer t_{out}, the lower the amount of disagreements in favour of positive agreements, at a price of an increased energy consumption.

and under Wi-Fi interference generated by a continuously active laptop, respectively. As the interference becomes heavier, the amount of positive agreements and the amount of disagreements drastically decrease after few iterations, hence the DPA ratio does not improve significantly. Our experiments therefore confirm our observations in Sect. II: packet-based n-way handshakes are not optimal under external interference.

C. 2-MAG: 2-way handshake enhanced with redundancy

To minimize the DPA ratio, we introduce redundancy of the last ACK packet as discussed in Sect. II-B, and we analyse the performance of a 2-way handshake in which the last ACK packet is sent k times, as illustrated in Fig. 2. For simplicity, in the remainder of this paper we will refer to this protocol as 2-MAG (2-way handshake Message-based AGreement).

Given the structure of JAG, a more fair comparison would involve a 3-way handshake message-based agreement protocol in which the last packet is sent k times. The choice of a 2-way handshake is driven by the results obtained in Fig. 9: a low n minimizes the probability of negative agreements, and therefore there are higher chances that 2-MAG sustains more positive agreements and outperforms JAG thanks to its redundant transmissions. We make sure to carry out a fair comparison by eliminating asymmetric links that would always lead to disagreements when using a two-way handshake.

In our implementation, hardware ACKs are enabled, i.e., the first ACK packet sent from $\mathcal{R}$ to $\mathcal{S}$ has a short and fixed-delay latency. Every other ACK packet will be generated via software by pre-loading the ACK into the radio buffer and by repeatedly sending its content k times. Please note that the preparation of the software ACK is time-critical, as one need to extract and analyse V before creating the ACK and loading it into the radio buffer.

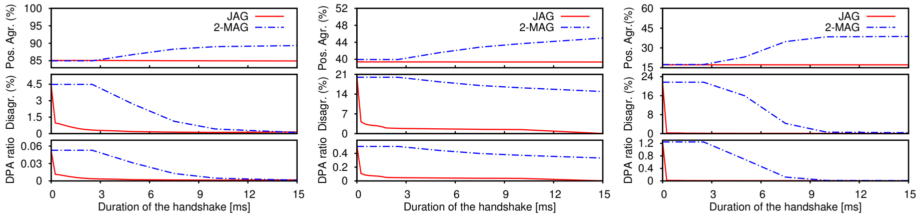
In order for $\mathcal{S}$ to consider V as successfully exchanged, it is sufficient to receive one ACK packet within a maximum waiting time t_{out}. Clearly, the longer t_{out}, the higher the likelihood that at least one ACK packet will be correctly decoded and the better 2-MAG will perform (at the price of an increased energy consumption). Hence, we compute t_{out} as the maximum time in which node $\mathcal{S}$ waits for a valid ACK packet from $\mathcal{R}$.

Fig. 10 shows the percentage of positive and negative agreements as well as disagreements obtained in the presence of interference using 2-MAG as a function of t_{out}. As expected, the longer t_{out}, the lower the amount of disagreements in favour of positive agreements. As this minimizes the DPA ratio, 2-MAG outperforms a generic n-way handshake without redundancy in the presence of external interference.

D. JAG: Jamming-based AGreement

We now evaluate the performance of JAG and compare it against 2-MAG. In particular, we are interested in comparing how the percentage of positive/negative agreements and disagreement change when we increase the duration of the handshake. Intuitively, the longer t_{out} for 2-MAG and the longer t_{jam} for JAG, the better the performance. However, it is important to see their distribution to study the protocols' energy-efficiency and their DPA ratio under interference.

Fig. 11 shows the results: JAG sustains a significantly lower amount of disagreements compared to 2-MAG already for small values of t_{jam}. For example, 2-MAG requires more than 7.5 ms to obtain less than 1% disagreement under Bluetooth interference, whereas JAG achieves this amount with a $t_{jam} \leq 250\mu$s.



(a) Bluetooth (b) Real Wi-Fi (c) Emulated Wi-Fi

Fig. 11. Compared to the 2-way handshake in which the last acknowledgment packet is sent k times, JAG performs better independent of the interfering source, as it reduces the duration of the handshake required to minimize the amount of disagreements.

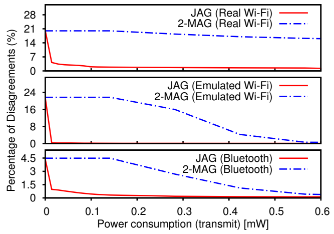
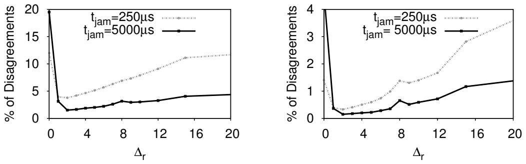


Fig. 12. Disagreements as function of energy for JAG and 2-MAG.



(a) Real Wi-Fi (b) Heavy Wi-Fi

Fig. 13. Role of Δ_r on the probability of disagreement.

Even though 2-MAG has a high number of positive agreements, it requires significantly higher values of t_{out} to reduce the amount of disagreements and the DPA ratio. JAG, instead, has a very low rate of disagreements under every type of interference even with small t_{jam}, which enables significant energy savings, as shown in Fig. 12. Furthermore, when t_{jam} is longer than the longest interference burst, we do not have any disagreements as discussed in Section IV-C. Obtaining this behaviour using packet-based approaches would require a significantly higher cost: Fig. 10(b) shows that even when sending bursts of ACKs for 100 ms, one cannot still guarantee the absence of disagreements. Hence, compared to packet-based approaches, JAG performs better and guarantees agreement with less costs and with weaker and more realistic assumptions about the underlying interference pattern.

Fig. 11(c) shows that the rate of disagreements obtained in the presence of emulated Wi-Fi interference tends to zero faster than the one obtained in the presence of real Wi-Fi interference. This is because the interference generated by JamLab contains fast transmissions with short idle and busy periods. Therefore, JAG has high chances to detect an idle period already when using a short t_{jam}.

In addition to t_{jam}, another parameter to be configured in JAG is Δ_r, which helps in compensating changes between r_s and the strength of the received jamming signal. Δ_r should be selected not too small (so to account for the inaccuracy of the RSSI readings), but at the same time not too large, as this would neutralize the benefits of having knowledge of r_s. Fig. 13 depicts the percentage of disagreements as a function of Δ_r: a value of 3 dBm offers a good trade-off.

Finally, we validate the goodness of JAG by running a long-term experiment in our second testbed deployed in a residential environment. In particular, we compare the performance of JAG and 2-MAG over time when using $t_{jam} = 500\mu$s for JAG and $t_{out} = 5ms$ for 2-MAG (Fig. 14). We do not change the configuration of the two protocols throughout the duration of the experiment. The interference in the environment changes significantly over the day: a lot of Wi-Fi activity was present during daytime in the weekend (May, 12-13), but it was quiet during night and on Monday (May, 14) during the day, as most people were not in their homes. Despite selecting a t_{out} 10 times higher than t_{jam}, JAG sustains a significantly lower amount of disagreements and outperforms 2-MAG during the whole duration of the experiment.

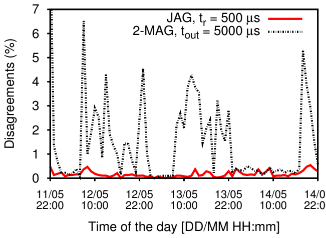
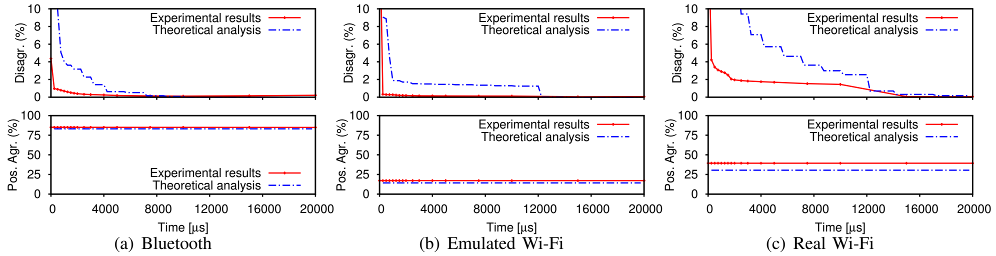


Fig. 14. Long-term experiment in a residential environment.

E. Predictability of JAG

We now evaluate the goodness of the probabilistic model presented in Sect. V with respect to the predictability of the performance of JAG. In order to do this, we firstly obtain the *pdf* of idle and busy periods using sensor nodes in wireless sniffer mode in the scenarios described in the previous sections, i.e., in the presence of JamLab's emulated interference and real Wi-Fi interference generated by a laptop (the *pdf*s in the presence of real Wi-Fi interference are shown in Fig. 5). Then, based on equation (2) and (3), we obtain an upper bound for the probability of obtaining disagreement and a lower bound for the probability of obtaining positive agreements as a function of t_{jam} using t_{pkt} = 1 ms, t_{ack} = 750 μs, and t = -90 dBm.



(a) Bluetooth (b) Emulated Wi-Fi (c) Real Wi-Fi

Fig. 15. Comparison of the rate of positive agreement and disagreement obtained running JAG on real wireless sensor nodes, and deriving the probabilities using the analytical model shown in Sect. V. The model actually returns a lower bound for positive agreements and an upper bound for disagreements.

By running JAG on real wireless sensor nodes, we verify experimentally whether the probabilistic model is able to predict the performance of JAG. The results illustrated in Fig. 15 show that our probabilistic model parametrizes correctly t_{jam} by giving an upper bound on the amount of disagreements and a lower bound on the amount of positive agreements, hence predicting the performance of the protocol correctly. Note that the probabilistic model was computed for every possible t_{jam}, whereas due to memory limitations of real nodes only a finite amount of t_{jam} were computed experimentally. Please note that Fig. 15 shows a different performance between emulated and real interference: whilst JamLab is designed to attain repeatability and test algorithms under the same conditions, real-world settings have several variables affecting their dynamics.

Based on our results, we can conclude that our theoretical model is indeed able to parametrize JAG and predict correctly the maximum amount of disagreements occurring for a given t_{jam}. This can be useful when the latter is shorter than the longest busy period created by interference (t_{busy}^{max}).

VII. Integration of JAG into MAC Protocols

As previously discussed, JAG is intended as a building block to construct protocols at different layers of the protocol stack. For example, it could be embedded into a MAC protocol to agree on the TDMA schedule or the next frequency channel. We now discuss how JAG can be integrated in existing MAC protocols to enhance their performance.

As many deployments gather environmental data and send them to a number of sinks, several convergecast MAC protocols have been proposed in sensor networks, such as Chrysso [16] and CoReDac [17]. In these protocols, nodes are logically organized into parent-children groups that may operate on different channels. In Chrysso [16], individual parent-children pairs collaboratively switch their communication channel as soon as performance degrades. In particular, a parent node monitors the average back-off time, and as soon as it exceeds a given threshold, it instructs all its children to carry out a channel switch by piggybacking the “switch-channel command” onto ACK messages, and then switches to the next channel. This operation is carried out for each parent-child pair individually, and can be considered a two-way handshake between child and parent (2-MAG) in which the information V to be agreed upon is contained in the second message. Please note that, on a high-level basis, V does not have to be necessarily included in the first message of the exchange: in a n-way handshake, V is in any case only used once the last message has been received, so it can be embedded in any of the messages exchanged in the handshake. The only difference with respect to 2-MAG is that, when piggybacking an information V into an ACK message, the latter cannot be sent as a hardware ACK as it contains extra-information.

JAG can be embedded into Chrysso by replacing the 2-way handshake between child and parent with a 3-way handshake in which the child sends an initial packet P, the parent answers with a software ACK containing the new channel to be used (V), and the child confirms the reception of V by jamming for a predefined amount of time t_{jam}. The parent node deems the exchange as successful (jamming sequence detected) or unsuccessful (jamming sequence not detected) depending on the results of a fast RSSI sampling, as described in Sect. IV-A.

The same principle can be used to enhance the performance of CoReDac [17], a TDMA-based convergecast protocol in which parent nodes split their reception slots into subslots, and assign one slot to each child in order to build a collection tree that guarantees collision-free radio traffic. As in Chrysso, also in CoReDac the assignment information used for synchronizing the TDMA-schedules is piggybacked onto ACK messages, and one can introduce a three-way handshake using JAG in the same way as described above. However, in the current version of CoReDac, there is a single aggregated ACK message containing the identifier of all children: this can be easily changed to individual ACKs to each child without affecting the overall protocol architecture.

The use of a 3-way handshake requires additional energy compared to the traditional message-based 2-way handshake implemented by Chrysso and CoReDac. However, this may pay off in the presence of interference, as it would increase the chances of agreement. As we have shown in our previous work [18], CoReDac performs poorly in the presence of interference, since when an ACK is lost, a sensor node needs to keep its radio on until it hears a new one, and integrating JAG may lead to substantial performance improvements.

VIII. Related Work

Agreement is a well-known problem in distributed systems. Pioneering work in the late 1970s highlighted the design challenges when attempting to coordinate an action by communicating over a faulty channel [2], [4].

In the context of wireless sensor networks, the agreement problem has not been widely addressed. The main focus has been on security for the exchange of cryptographic keys [19], and on average consensus for nodes to agree on a common global value after some iterations [20]. Similarly to these studies, our work aims at protocols that allow a set of nodes to agree on a piece of information. In addition, we also tackle agreement under interference and provide a lightweight energy-efficient solution that fits applications with strict performance requirements.

Our work is motivated by studies reporting the degrading QoS caused by the overcrowding of the RF spectrum in unlicensed bands [3]. Several solutions have been proposed: Chowdhury and Akyildiz identify the type of interferer and schedule transmissions accordingly [21]. Liang et al. increase the resilience of packets challenged by Wi-Fi interference using multi-headers and FEC techniques [11]. Other protocols, such as Chrysso and ARCH, dynamically switch the communication frequency as soon as interference is detected [22], [16]. As these protocols rely on packet exchanges to coordinate the channel switching, one can use JAG to improve their performance, as discussed in Sect. VII.

Another set of studies propose to cope with interference by exploiting its idle or busy periods. Noda et al. have proposed a channel quality metric based on the availability of the channel over time, which quantifies spectrum usage [23]. Hauer et al. report the interference observed by a mobile body area network in public spaces, and the study shows the intermittent interference caused by Wi-Fi AP in all IEEE 802.15.4 channels [7]. Similarly, Huang et al. have shown that Wi-Fi traffic inherently leaves "a significant amount of white spaces" between 802.11 frames [24]. BurstProbe uses a probing mechanism to periodically measure burst error patterns of all links used in the deployment and, whenever the interference patterns leave predicted bounds, a warning is issued so that one can reconfigure the deployed network [25]. Similarly to these studies, JAG exploits idle times for data packets, but also leverages the bursty nature of interfering sources to achieve reliable agreements through the use of jamming signals.

IX. Conclusions

In this paper, we propose JAG, a simple and efficient agreement protocol for wireless sensor networks exposed to external interference. JAG introduces a novel technique that utilizes jamming signals to acknowledge the reception of a packet. Our results show that JAG outperforms traditional methods using packet-based acknowledgements. Further, JAG provides predictable performance in that it keeps, within a specified energy budget and delay time, the probability of disagreements below a pre-defined threshold even in the presence of external interference, and in that it can be configured to always reach agreement (positive or negative) in a finite amount of time.

A limitation of the current version of JAG is that jamming sequences do not provide identity information, and hence may be generated by a malicious device. JAG partially solves the problem by using a mechanism to verify that the strength of the jamming signal equals the one that would be produced by the device of interest. However, security is an important concern nowadays, and it would be important to unequivocally guarantee the identity of the jamming node by means of authentication. We will address this issue in future work.

Acknowledgments

This work has been partially supported by the European Commission with contracts FP7-2007-2-224053 (CONET, the Cooperating Objects Network of Excellence), and INFSO-ICT-317826 (RELYonIT). This research has been also partially financed by VINNOVA, the Swedish Agency for Innovation Systems, by the DFG-funded Cluster of Excellence 306/1 "Inflammation at Interfaces", and by the Mercatur Research Center Ruhr Grant number Pr-2011-0014 (SEVERE).

References

[1] T. Abdelzaher et al., "EnviroTrack: Towards an Environmental Computing Paradigm for Distributed Sensor Networks," in *24th ICDCS*, 2004.

[2] E. Akkoyunlu et al., "Some Constraints and Tradeoffs in the Design of Network Communications," in *Symp. on Op. Syst. Princ. (SOSP)*, 1975.

[3] G. Zhou, J. Stankovic, and S. Son, "Crowded Spectrum in Wireless Sensor Networks," in *3rd Worksh. on Emb. Net. Sens. EmNetS*, 2006.

[4] J. Gray, "Notes on Data Base Operating Systems," in *Operating Systems, an Advanced Course, pp. 393–481*, 1978.

[5] R. Musaloiu-E. and A. Terzis, "Minimising the Effect of WiFi Interference in 802.15.4 WSN," *IJSNet*, vol. 3, no. 1, pp. 43–54, Dec. 2007.

[6] C.A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation," in *10th ACM/IEEE IPSN*, 2011.

[7] J. Hauer, V. Handziski, and A. Wolisz, "Experimental Study of the Impact of WLAN Interf. on IEEE 802.15.4 BANs," in *6th EWSN*, 2009.

[8] J. Hauer, A. Willig, and A. Wolisz, "Mitigating the Effects of RF Interference through RSSI-Based Error Recovery," in *7th EWSN*, 2010.

[9] A. Kamerman and N. Erkocevic, "Microwave Oven Interf. on Wireless LANs Operating in the 2.4 GHz ISM Band," in *IEEE PIRMC'97*.

[10] W.-B. Pöttner, S. Schildt, D. Meyer, and L. Wolf, "Piggy-Backing Link Quality Measurements to IEEE 802.15.4 ACKs," in *8th MASS*, 2011.

[11] C. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi Interf. in Low Power ZigBee Networks," in *8th ACM SenSys*, 2010.

[12] K. Jamieson and H. Balakrishnan, "PPR: Partial Packet Recovery for Wireless Networks," in *ACM SIGCOMM*, 2007.

[13] C.A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig, "Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks," in *4th IEEE SenseApp*, 2009.

[14] J. Polastre, J. Hill, and D. Culler, "Versatile Low-Power Media Access for Wireless Sensor Networks," in *2nd ACM SenSys*, 2004.

[15] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki: a Lightweight and Flexible Op. System for Tiny Networked Sensors," in *1st EmNetS*, 2004.

[16] V. Iyer, M. Woehrle, and K. Langendoen, "Chrysso: A Multi-channel Approach to Mitigate External Interference," in *8th IEEE SECON*, 2011.

[17] T. Voigt and F. Österlind, "CoReDac: Collision-Free Command-Response Data Collection," in *13th IEEE ETFA*, 2008.

[18] C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M.A. Zúñiga, "Making Sensornet MAC Protocols Robust Against Interference," in *7th EWSN*, 2010.

[19] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A Key Management Scheme for WSN using Depl. Knowledge," in *23rd INFOCOM*, 2004.

[20] L. Xiao, S. Boyd, and S. Lall, "A Scheme for Robust Distributed Sensor Fusion based on Average Consensus," in *4th ACM/IEEE IPSN*, 2005.

[21] K. Chowdhury and I. Akyildiz, "Interferer Classification, Channel Selection and Transmission Adaptation for WSN," in *IEEE ICC*, 2009.

[22] M. Sha, G. Hackmann, and C. Lu, "ARCH: Practical Channel Hopping for Reliable Home-Area Sensor Networks," in *17th IEEE RTAS*, 2011.

[23] C. Noda et al., "Quantifying the Channel Quality for Interference-Aware Wireless Sensor Networks," *ACM SIGBED Review*, vol. 8, no. 4, 2011.

[24] G. Huang et al., "Beyond Co-Existence: Exploiting WiFi White Space for Zigbee Performance Assurance," in *18th IEEE ICNP*, 2010.

[25] J. Brown et al., "BurstProbe: Debugging Time-Critical Data Delivery in Wireless Sensor Networks," in *8th EWSN*, 2011.

Paper E

C.A. Boano, H. Wennerström, M.A. Zúñiga, J. Brown, C. Keppitiyagama, F.J. Oppermann, U. Roedig, L.-Å. Nordén, T. Voigt, and K. Römer. **Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers.** *In Proceedings of the 5th Extreme Conference on Communication (ExtremeCom).* Thórsmörk, Iceland. August 2013. **Best Paper Award.**

Summary. This paper studies the impact of temperature on various sensornet platforms systematically, and characterizes the dependency between link quality and temperature variations. First, the paper analyses the impact of temperature on several links in a one-year long outdoor deployment in Sweden, highlighting a significant attenuation in received signal strength at high temperatures. Second, by means of controlled testbed experiments, this paper shows the different impact on transmitting and receiving nodes, and highlights how the attenuation in received signal strength follows a similar linear trend in all sensornet platforms. The paper captures the dependency between link quality and temperature variations in a simple first-order model that can be used to predict the performance of a network considering the particular temperature profile of a given environment.

My contributions. I am the main author of this paper and was responsible for the systematic characterization of the dependency between link quality and temperature variations. I wrote the vast majority of the paper in collaboration and discussion with the co-authors, and carried out the experiments in the Section 3.2. The analysis of the impact of temperature in the outdoor deployment (Section 3.1) was carried out by Hjalmar Wennerström, and the model-based temperature profile section (Section 4.2) has been written mostly by Marco Zuniga. I presented the paper at ExtremeCom'13.

©2013 ACM.
ISBN: 978-1-4503-2171-6.
(Did not yet appear in the ACM Digital Library)

- On the author's institutional repository;
- In any repository legally mandated by the agency funding the research on which the work is based.

Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers

Carlo Alberto Boano[†][*], Hjalmar Wennerström[§], Marco Antonio Zúñiga[¶],
James Brown[‡], Chamath Keppitiyagama[♮], Felix Jonathan Oppermann[†],
Utz Roedig[‡], Lars-Åke Nordén[§], Thiemo Voigt[♮§], and Kay Römer[†]

[†]Institute of Computer Engineering
University of Lübeck, Germany

[‡]School of Computing and Communications
Lancaster University, United Kingdom

[¶]Embedded System Group
TU Delft, The Netherlands

[§]Department of Information Technology
Uppsala University, Sweden

[♮]Swedish Institute of Computer Science
Stockholm, Sweden

ABSTRACT

Temperature is known to have a significant effect on the performance of radio transceivers: the higher the temperature, the lower the quality of links. Analysing this effect is particularly important in sensor networks because several applications are exposed to harsh environmental conditions. Daily or hourly changes in temperature can dramatically reduce the throughput, increase the delay, or even lead to network partitions. A few studies have quantified the impact of temperature on low-power wireless links, but only for a limited temperature range and on a single radio transceiver. Building on top of these preliminary observations, we design a low-cost experimental infrastructure to vary the on-board temperature of sensor nodes in a repeatable fashion, and we study systematically the impact of temperature on various sensornet platforms. We show that temperature affects transmitting and receiving nodes differently, and that all platforms follow a similar trend that can be captured in a simple first-order model. This work represents an initial stepping stone aimed at predicting the performance of a network considering the particular temperature profile of a given environment.

Categories and Subject Descriptors

Computer Systems Organization [**Embedded and cyber-physical systems**]: Sensor networks.

Keywords

Signal strength, Temperature, Wireless Sensor Networks.

1. INTRODUCTION

Wireless sensor networks (WSNs) have proven to be an excellent monitoring tool and nowadays many installations exist. They are, for example, used to monitor natural phenomena such as glaciers, infrastructures such as bridges, or production processes on oil platforms. Many of these deployments are heavily exposed to the environment and experience extreme temperature changes within a day and over seasons. Temperature has a significant impact on wireless communication and a system has to be designed to handle all possible temperature changes over the deployment lifetime. This is of particular importance if we rely on the system and expect a deterministic performance at any given point in time. For example, we expect that a WSN-based process automation on an oil rig operates reliably while the installation is cycling through the extreme temperature changes that are typically found in such deployments. A system failure caused by a wrong prediction of the impact of temperature changes on wireless communication is not acceptable.

Many studies describing experiences from WSN outdoor deployments have reported that diurnal (day/night) and seasonal (summer/winter) fluctuations of ambient temperature have a strong impact on communication quality. Lin et al. [1] have found a daily variation in the received signal strength (RSS) of up to 6 dBm, with the highest RSS values being recorded during night-time. Similarly, in their deployment in an Australian outdoor park, Sun and Cardell-Oliver [2] have measured on-board temperature daily variations between 10 and 50 °C, and noticed that links perform very differently between day and night. Also Thelen et al. [3] have noticed a drastic decrease of RSS at high temperatures in their potato-field deployment.

While the *macro-view* of the problem is clear (temperature has an effect on signal strength and link quality), this knowledge does not help us to fully understand the dependency between link quality and temperature. Furthermore, existing work does not allow us to predict the performance of a network with respect to communication-related temperature dependencies. The aim of this work is hence to develop a *micro-view* of the problem by analysing systematically the impact of temperature on different radio transceivers. We design a low-cost experimental infrastructure to vary the on-board temperature of nodes in a repeatable fashion and study the effects on transmitting and receiving nodes, iso-

[*]Contact author. E-mail: cboano@iti.uni-luebeck.de.
The research leading to these results has received funding from the European Union 7th Framework Programme (FP7-ICT-2011-8) under grant agreement n° 317826 (RELYonIT).

ExtremeCom '13, August 24-30, 2013, Thorsmork, Iceland.
Copyright 2013 ACM 978-1-4503-2171-6 ...$15.00.

lating hardware-specific effects. Our results show that all platforms follow a similar trend that can be captured in a relatively simple first-order generic model for low-power wireless transceivers. Such a model can be used for planning and constructing wireless sensor networks providing dependable service despite temperature changes.

In the next section, we describe existing work in the outlined research area. In Sect. 3 we present results from a 1-year long outdoor deployment in Sweden that we used as a starting point for this work. We then describe and analyse the results of extensive lab experiments to systematically study the effects of temperature in a controlled setting. We develop a first-order model of temperature and link quality dependency in Sect. 4 and conclude our paper in Sect. 5.

2. RELATED WORK

Results by Bannister et al. [4] from an outdoor deployment and from experiments in controlled scenarios have revealed that an increase in temperature causes a reduction in RSS. In their experiments in a climate chamber, the authors observe a linear decrease in RSS of about 8 dB over the temperature range 25-65 °C and show that this reduction may have severe consequences on the connectivity of a network. These results were confirmed by experiments by Boano et al. [5], [6], showing that one can safely decrease the transmission power of communications at low temperatures without deteriorating the performance of the network.

A recent long-term outdoor deployment by Wennerström et al. [7] has further shown that the average packet reception rate (PRR) in a WSN of 16 Tmote Sky nodes dropped by more than 30% when changing temperature from -5 to 25 °C, and that a clear degradation in PRR and average link quality occurred during summer, confirming that daily and seasonal fluctuations of ambient temperature have a strong impact on the quality of sensornet communications.

These existing works simply report the degradation of signal strength and link quality as a consequence of an increase in ambient temperature and do not provide a deeper analysis of the problem. In addition, every reported analysis is unique in terms of experimental setup and hardware. The used radio chips range from Nordic NRF903 [2] and CC1000 [3] to the popular CC1020 [6] and CC2420 transceivers [1], [7], making it difficult to separate general from hardware-specific effects.

Bannister et al. [4] have attempted to quantify the loss of RSS due to temperature changes, but only for a limited temperature range and for a single radio chip. Furthermore, when simulating the reduction of communication range and connectivity degradation due to an increase in ambient temperature, the authors assume that communicating nodes have similar temperatures.

This work goes beyond existing work and studies the impact of sender and receiver temperature on link quality systematically using different hardware platforms. After isolating hardware-specific effects, we show that temperature affects all platforms in a similar way and derive a model that captures its impact on low-power wireless transceivers.

3. EXPERIMENTAL RESULTS

In order to get a deeper understanding of the impact of temperature on WSNs, we study the evolution of link quality over one year in an outdoor deployment in Sweden. Our analysis shows that temperature has a strong impact on communication, with visible daily and seasonal differences.

Building on top of these results, we carry out a large set of experiments in controlled settings, where we can repeat and alter the conditions at different nodes separately. In all our experiments, we analyse the impact of temperature by measuring the hardware-based link quality metrics in IEEE 802.15.4 compliant radio transceivers [8], namely the received signal strength indicator upon packet reception (RSSI) and in absence of packet transmissions (noise floor), and the link quality indicator (LQI)[1].

3.1 Long-Term Outdoor Deployment

We now describe the impact of temperature on communication that we have observed in our outdoor deployment at a Swedish meteorological station spanning over a whole year.

Experimental Setup. We have deployed a sensor network comprising 16 TelosB sensor nodes outside Uppsala, Sweden, in an open field isolated from human activity and absence of electromagnetic interference. Sensor nodes are mounted on poles along a 80 meter straight line at intervals of 0, 20, 40, and 80 meters: on each pole, two nodes are mounted at 0.5 and 1.5 meters height, respectively. The nodes are powered via USB and attached to a Sensei-UU testbed [9], ensuring reliable and continuous data logging.

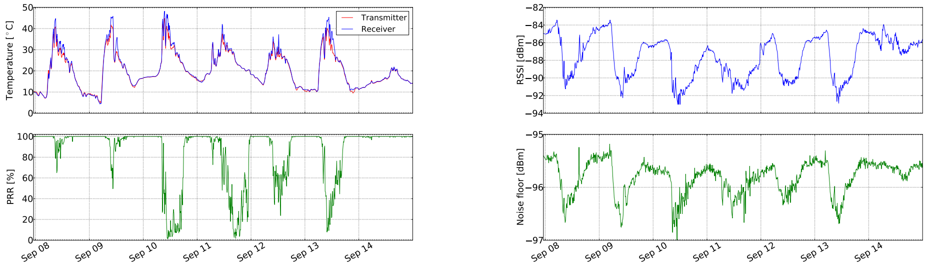
The software running on the sensor nodes periodically sends packets between every possible pair of nodes and works as follows. Each node is assigned the sender-role in a round-robin fashion every 30 seconds. During this phase, the designated sender transmits one packet per second addressed to each of the other nodes, again in a round-robin manner. When a packet is received by the intended recipient, a response packet addressed to the sender is sent. Each time a sensor node receives a packet – including when it is not the intended recipient – it logs several statistics about the received packet, namely RSSI, LQI, and noise floor. Onboard ambient temperature is measured on each node every two seconds using the on-board SHT11 temperature sensor. More details on the experimental setup can be found in [7].

Impact of temperature on PRR. To highlight the impact that ambient temperature has on the links deployed in our outdoor WSN, we focus on a specific link, close to the edge of the communication range. Fig. 1(a) (top) shows the temperature of two nodes (transmitter and receiver) forming a unidirectional link during a week in September. Temperature varies as much as 40 °C between day and night since sensor nodes are enclosed into air-tight enclosures and exposed to direct sunlight. Therefore daily temperature fluctuations may cause a combined overall variation between the two nodes of up to 80 °C. Although the highest variations occur over the 24-hours, temperature can fluctuate by as much as 34.9 °C within one hour, as we show in Table 1, in which we summarize the largest temperature ranges observed in our 12-months deployment for different time intervals.

Fig. 1(a) (bottom) further shows that each substantial increase in temperature (typically occurring during daytime) results in a decrease in PRR, leading to an almost complete disruption of the connectivity between the two nodes.

Impact of temperature on RSSI and noise floor. The decrease in PRR is strongly correlated with a decrease

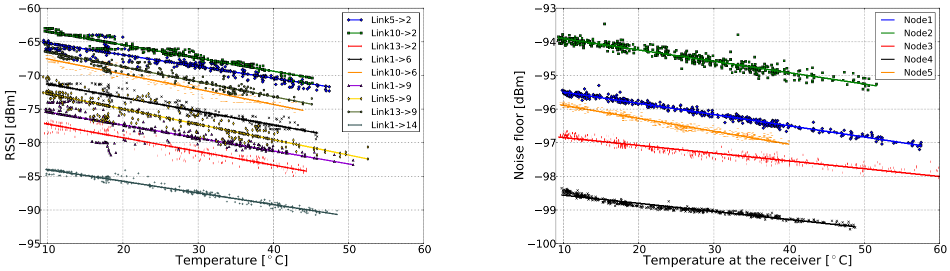
[1] Please notice that the RSSI readings from all sensor nodes employed in our experiments are uncalibrated.



(a) Temperature and PRR

(b) RSSI and Noise floor

Figure 1: Temperature has a strong impact on the quality of links in our outdoor WSN. During daytime, when temperature is high, there is a significant reduction in PRR (a). Also the trend of RSSI and noise floor resembles the one of temperature, with a sharp decrease when temperature increases (b).



(a) RSSI

(b) Noise floor

Figure 2: The relationship between RSSI and temperature (a) and between noise floor and temperature (b) can be approximated as a linear function, and the trend is similar for different nodes.

	1 year	1 month	1 day	1 hour
Lowest temp. (°C)	-22.2	-3.0	7.2	21.2
Highest temp. (°C)	61.3	63.7	63.8	55.9
Temp. difference	82.5	66.7	56.6	34.9

Table 1: Largest temperature variations on a single node as seen in our outdoor deployment.

in the RSSI computed over the received packets, as shown in Fig. 1(b) (top), hinting that the change in temperature – and not external interference – was the cause of the packet loss. In particular, the RSSI fluctuates between -84 and -92 dBm, the latter being the threshold below which no packets are received. Interestingly, also the noise floor follows a trend similar to the RSSI and decreases as temperature increases, but to a much lower extent, as shown in Fig. 1(b) (bottom).

The strong correlation between temperature, RSSI, and noise floor is highlighted in Fig. 2(a) and 2(b), respectively. Fig. 2(a) shows the RSSI and the combined temperature of sender and receiver for nine links with different link quality over a timespan of three days. The relationship between temperature and RSSI can be approximated as a linear function and is clearly visible despite the intrinsic noise produced by long-term measurements. Using linear regression we have observed that different links have a similar trend, with an average slope of -0.205 and a standard deviation of 0.026.

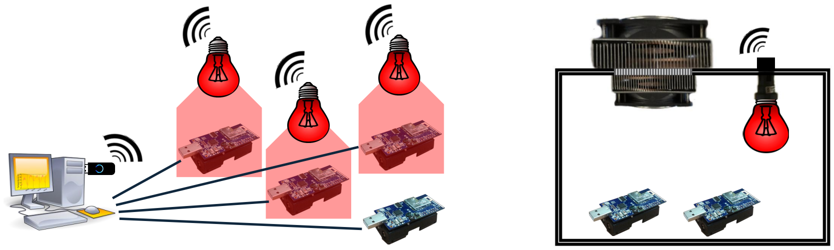
Fig. 2(b) shows the noise floor of five nodes over the same 3 days. Also in this case, the relationship with temperature is approximately linear, with a similar slope among different nodes, but with a less pronounced decrease compared to RSSI (average slope of -0.034 ± 0.006).

3.2 Controlled Testbed Experiments

To get a deeper understanding of the effects observed in Sect. 3.1, we have augmented an existing sensornet testbed with the ability of varying the on-board temperature of sensor motes and *reproduce the impact of temperature on link quality in a repeatable fashion.* We use this low-cost testbed infrastructure to systematically study the impact of temperature on different hardware platforms and to isolate the effects of temperature on transmitting and receiving nodes.

Experimental Setup. Fig. 3(a) shows an overview of our controlled experimental setup. We have extended an existing WSN testbed with the ability of varying the on-board temperature of sensor motes in the range -5 to +80 °C using infrared light bulbs placed on top of each sensor node. The light bulbs can be remotely dimmed using the 868 MHz frequency, and hence their operations do not interfere with the communications between the wireless sensor nodes, as the latter use the 2.4 GHz ISM band. In order to cool down the motes below room temperature, we have built custom Polystyrene enclosures as shown in Fig. 3(b), in which, in addition to the light bulb, a Peltier air-to-air assembly module by Custom Thermoelectric cools the temperature down to -5 °C when the enclosure is kept at room temperature and the light bulb is off. As we only have a limited number of Peltier enclosures, some of the nodes in the testbed are only warmed by the infrared light bulbs between room temperature and their maximum operating temperature range.

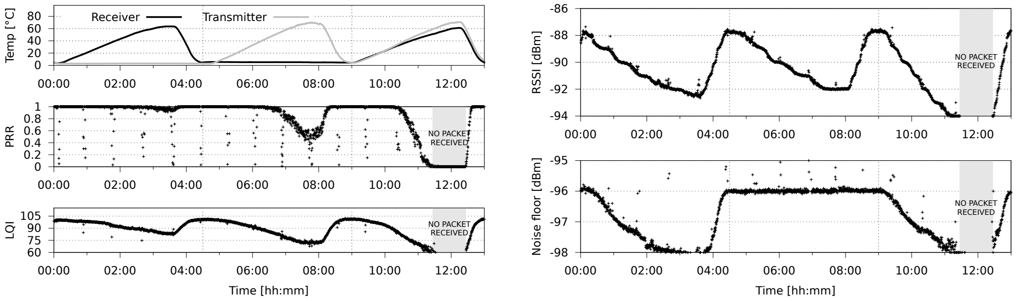
Our testbed is composed of Maxfor MTM-CM5000MSP and Zolertia Z1 nodes employing the CC2420 radio [10], as



(a) Setup overview

(b) Sketch of a Peltier enclosure

Figure 3: Experimental setup in controlled testbed experiments.



(a) PRR and LQI

(b) RSSI and Noise floor

Figure 4: Impact of temperature on the quality of links in our controlled testbed. We heat transmitter and receiver nodes separately first, and then both of them at the same time. When temperature increases, PRR, LQI, and RSSI decrease significantly, with the highest impact occurring when both nodes are heated at the same time. The periodic noise is due to a Wi-Fi access point beaconing in proximity of the testbed.

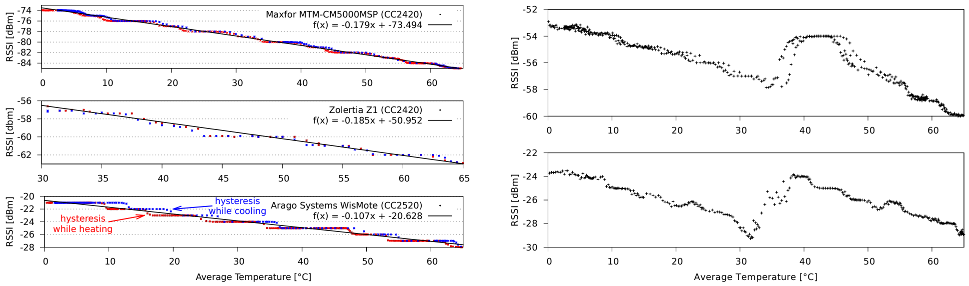
well as of Arago Systems WisMotes employing the CC2520 transceiver [11]. Sensor nodes are divided in pairs and form bidirectional links operating on different physical channels to avoid internal interference. All sensor nodes run the same Contiki software: each sensor node continuously measures the ambient temperature and relative humidity using the on-board SHT11 or SHT71 digital sensors, and periodically sends packets to its intended receiver at a speed of 128 packets per second using different transmission power levels. Statistics about the received packets are logged using the USB backchannel and are available remotely.

Validation of our controlled setup. Using our controlled testbed setup, we are able to reproduce the impact of temperature on link quality in a very fine-grained way. In a first experiment using Maxfor nodes, every link in the testbed is exposed to three heat cycles. First, each individual node, i.e., first the transmitter and then the receiver, is heated from 0 up to 65 °C. Afterwards, both nodes are heated in the same temperature range at the same time. Fig. 4(a) illustrates the impact of temperature on PRR and LQI on a particular link. The evolution of temperature at the transmitter and at the receiver over the 13-hours experiment is shown in the top figure. In correspondence to each increase of temperature, PRR and LQI decrease significantly, with the highest impact occurring when both nodes are heated. With both nodes heated, indeed, no packet was received and the connectivity between the two nodes was interrupted until the temperature started to decrease. Fig. 4(a) also shows that the packet loss rate is more pronounced when the transmitter is heated compared to the case in which only the receiver is heated, something that we have observed in the majority of links in our testbed.

Fig. 4(b) illustrates the impact of temperature on RSSI (top figure) and noise floor (bottom figure). The RSSI decreases in a similar way when transmitter and receiver are heated separately, whereas the decrease is more pronounced if both transmitter and receiver are heated at the same time. This proves that temperature decreases both the transmitted and received power [4], whereas the noise floor only decreases when the receiver node is heated, with an absolute variation smaller than the one of RSSI.

These results hence prove the validity of our setup and confirm the measurements obtained in our outdoor deployment, quantifying precisely the impact on temperature on each individual node. We now derive a set of observations obtained running experiments using the same experimental setup, i.e., three heat cycles in which each node is heated individually first and then both nodes are heated at the same time, on different hardware platforms.

The decrease in RSSI is consistent among different platforms. The trend observed in our outdoor deployment showing that RSSI decreases in an approximately linear fashion with temperature holds for different platforms and different radio chips, but with a different slope. Fig. 5(a) shows the relationship between RSSI and temperature obtained on different platforms when heating both nodes at the same time. The hardware platforms employing the same CC2420 radio exhibit approximately the same slope.



(a) Loss in RSSI when temperature changes

(b) Non-linearities in the CC2420 radio

Figure 5: Figure (a) shows that the relationship between RSSI and temperature is similar when using different hardware platform and can be approximated as a linear function, but with different parameters. Figure (b) shows the non-linearities in the response of the CC2420 radio measured using Maxfor nodes. Temperature on the x-axis is computed as the average temperature of the transmitter and receiver temperature.

The decrease in RSSI does not depend on how quickly temperature changes. In our setup, the heat cycles are characterized by a slow increase in temperature followed by a quicker cooling phase, as can be seen in Fig. 4(a). This allows us to observe that both RSSI and noise floor are not affected by how quickly temperature varies. Hence, the impact of temperature can be modelled using the absolute temperature value at the transmitter and receiver nodes.

Discrete steps. On close inspection in Fig. 5(a), one can observe discrete steps in the relationship between RSSI and temperature. For the CC2420 platforms, the size of the prominent steps is 2 dBm, whereas for platforms employing the CC2520 radio the step is 1 dBm large. Bannister [12] has attributed the loss of RSSI to the loss of gain in the CC2420 Low Noise Amplifier (LNA). Our experiments bring further evidence to strengthen this claim, as there are references to 2 dBm steps in the CC2420 datasheet [10] with regard to the operation of the Automatic Gain Controller (AGC).

Hysteresis. Fig. 5(a) also shows an hysteresis in the relationship between RSSI and temperature that can be seen comparing the RSSI curve obtained when heating and when cooling down the motes. As for the discrete steps, the hysteresis also can be attributed to the operation of the AGC in the CC2420 radio. According to the CC2420 datasheet, hysteresis on the switching between different RF front-end gain modes is set to 2 dBm [10].

Non-linearity in the CC2420 curve. In our experiments, we have also noticed visible non-linearities when the RSSI is ≈ -28 and -58 dBm in the CC2420 platform, as shown in Fig. 5(b). These non-linearities were also measured by Chen and Terzis [13], and may lead to a false approximation in case the RSSI of the considered link falls *exactly* in this region (as in the experiments of [4]). When deriving our linear approximation for the CC2420 transceiver, we hence do not consider links falling in this range.

RSSI loss on transmitter and receiver. Fig. 6(a) shows the relationship between RSSI and temperature obtained on Maxfor nodes when transmitter and receiver nodes are heated individually and when both nodes are heated at the same time. Top and bottom figures refer to the same link, but are obtained using a different transmission power. Despite the link is the same, the relationship between RSSI and temperature is slightly different, with a steeper decrease when the receiver is heated in the top figure. Although a comparison between curves is difficult due to the AGC operations (depending on whether we capture the transition between two discrete steps, we may obtain slightly different slopes), by averaging the data from all our experiments we have obtained a relationship between receiver and transmitter of 0.5348 ± 0.061. The RSSI seems hence to have a slightly steeper slope when the receiver node is heated.

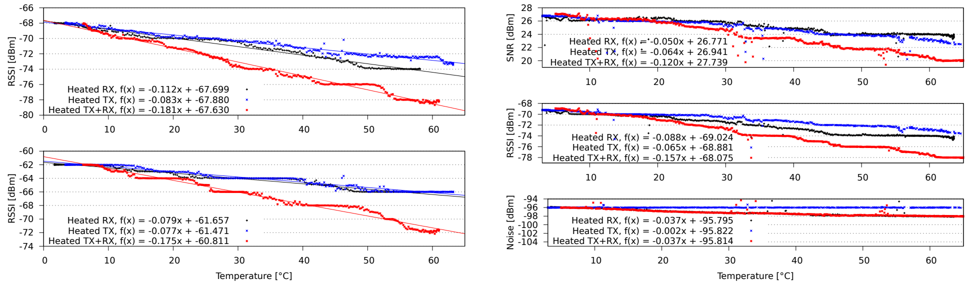
Impact on noise floor and SNR. Fig. 6(b) illustrates how noise floor, RSSI, and signal to noise ratio (SNR) vary on a given link when transmitter and receiver nodes are heated individually and at the same time. Since the noise floor decreases only when the receiver is heated, an increase in temperature on the transmitter has an higher impact on the SNR compared to an increase in temperature at the receiver. This also explains the different impact in PRR when heating the nodes individually that we observed in Fig. 4(a).

4. PLATFORM MODELS

The effect of temperature on electric conductors and semiconductors is well known. Various models have been created for a large range of devices to capture the relation between ambient temperature and electric conductance (and current leakage). Our goal is to build on top of this knowledge to create a generic model for low-power radio transceivers. It is important to remark that the goal of our model is not to benchmark a specific radio chip against others, as this is already done by manufacturers. Our goal is to develop a simple model to predict the performance of a network under extreme environmental settings. We now describe the overarching effect of temperature on radio transceivers and derive a generic model for low-power wireless transceivers.

4.1 The effect of temperature on RSS

In electric conductors, a higher temperature increases the resistance of the medium, whereas in semiconductors it leads to current leakages. In practice this means that, for a given voltage, a higher temperature reduces the current and hence the power of a device. In radio transceivers, these phenomena imply that a raise in temperature will reduce the SNR. A decrease in SNR leads to a lower link quality and a shorter radio link, which in turn may lead to lower throughput, higher delay or even network partitioning. Hence, our goal is to model the effect of temperature on SNR. Denoting PL as the path loss between a transmitter-receiver pair, P_t as the transmission power, P_r as the received power, and P_n



(a) Loss in RSSI when using different TX powers (b) Loss in noise floor, RSSI, and SNR for a given link

Figure 6: Relationship between RSSI, noise floor, SNR and temperature when transmitter (blue) and receiver (black) nodes are heated individually, and when both nodes (red) are heated at the same time.

as the noise floor at the receiver, the SNR is known to be:

$$\begin{aligned} SNR(dB) &= P_t - PL - P_n \\ &= (P_t - P_n) - (P_t - P_r) \end{aligned} \tag{1}$$

As we have shown in our empirical measurements, an increasing temperature has 3 main effects on the signal strength of radio transmissions; it (i) decreases the transmitted power, (ii) decreases the received power, and (iii) decreases the noise floor. We now model these three effects in Eq. 1.

4.2 A first-order model

Denoting α, β, γ as constants with units dB/K, and T_t, T_r as the temperature in Kelvin of transmitter and receiver, the effect of temperature on SNR can be defined as:

$$\begin{aligned} SNR &= (P_t - \alpha\Delta T_t) - (PL + \beta\Delta T_r) \\ &\quad -(P_n - \gamma\Delta T_r + 10\log_{10}(1+\tfrac{\Delta T_r}{T_r})) \\ &= P_t - PL - P_n - \alpha\Delta T_t \\ &\quad -(\beta-\gamma)\Delta T_r - 10\log_{10}(1+\tfrac{\Delta T_r}{T_r}) \end{aligned} \tag{2}$$

The proportional relation between ΔT and the constants α (effect on transmitted power), β (effect on received power) and γ (effect on noise floor) is based on the empirical observations made in the previous sections. The term $10\log_{10}(1+\frac{\Delta T_r}{T_r})$ is derived analytically from the well-known thermal equation. There are two important trends to highlight in this model. First, changes in temperature have a higher impact on the transmitted and received powers (linear relation of α and β), than on the thermal noise (logarithmic relation). Second, to some extent it is counter-intuitive that a higher temperature decreases the noise floor (negative sign of γ). This effect was also observed by Bannister, and he hypothesizes that it is due to the losses in the signal amplifier [12]. That is, a higher temperature not only reduces the gain of the signal but also the gain of the noise, and hence, the received signal strength (RSSI) is lower for both.

The accuracy of our model depends on identifying the right values for α, β and γ. In our case, these parameters are given by the slopes of the linear trends observed in our empirical results. These parameters are platform dependant, and hence require a systematic and fine-grained evaluation. Our testbed was designed to accomplish exactly that. For example, a network manager willing to deploy a network using the Maxfor platform, can use the slopes obtained in Fig. 6(b): $\alpha = 0.065$, $\beta = 0.088$ and $\gamma = 0.037$. Assuming that the network will be deployed in an environment where the maximum and minimum day temperature are 50 and 5°C respectively, the network manager can predict that the links can suffer an attenuation of $(\alpha + \beta - \gamma)\Delta T = 5.22$ dB (5 dB according to the SNR measurements in Figure 6(b) top). This level of attenuation can easily push a good link (with 100% PRR) to have a PRR of 0%.

5. SUMMARY AND OUTLOOK

The central tenet of our study is that the important role played by ambient temperature in the performance of sensor networks can (and must) be analysed in a systematic way. Motivated by initial studies focusing on single platforms, we use a low-cost yet precise testbed to show that most platforms have similar intrinsic characteristics that can be easily modelled. Our results capture with good accuracy how temperature affects the signal strength in transmitters and receivers. A thorough understanding of the effect of temperature on low-power wireless links is a first necessary step of a much broader goal: the ability to predict the performance of sensor networks in various environmental settings.

6. REFERENCES

[1] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proc. of the 4th SenSys*, 2006.

[2] J. Sun and R. Cardell-Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *Proc. of the 2th RealWSN*, 2006.

[3] J. Thelen et al. Radio wave propagation in potato fields. In *Proc. of the 1st WiNMee*, 2005.

[4] K. Bannister, G. Giorgetti, and S. K. S. Gupta. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proc. of the 5th HotEmNets*, 2008.

[5] C.A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. The impact of temperature on outdoor industrial sensornet applications. *IEEE TII*, 6(3), 2010.

[6] C.A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proc. of the* 1^{st} *Sensappeal*, 2009.

[7] H. Wennerström et al. A long-term study of correlations between meteorological conditions and 802.15.4 link performance. In *Proc. of the 10th SECON*, 2013.

[8] N. Baccour et al. Radio link quality estimation in wireless sensor networks: a survey. *TOSN*, 8(4), 2012.

[9] O. Rensfelt et al. Sensei-UU: a relocatable sensor network testbed. In *Proc. of the 5th WiNTECH*, 2010.

[10] Texas Instr. *CC2420 datasheet, revision SWRS041c*, 2013.

[11] Texas Instr. *CC2520 datasheet, revision SWRS068*, 2007.

[12] K. Bannister. Impacts of thermal reduction in transceiver performance on outdoor sensing networks. Master's thesis, Arizona State University, Phoenix, AZ, USA, 2009.

[13] Y. Chen and A. Terzis. On the mechanisms and effects of calibrating RSSI measurements for 802.15.4 radios. In *Proc. of the 7th EWSN*, 2010.

Paper F

C.A. Boano, M.A. Zúñiga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Römer. **TempLab: A Testbed Infrastructure to Study the Impact of Temperature on Wireless Sensor Networks.** *In Proceedings of the 13^{th} ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* Berlin, Germany. April 2014.

Summary. This paper presents TempLab, an extension for wireless sensor network testbeds that allows to control the on-board temperature of sensor nodes and to study the effects of temperature variations on the network performance in a precise and repeatable fashion. TempLab can accurately reproduce traces recorded in outdoor environments with fine granularity, while minimizing the hardware costs and configuration overhead. Hence, TempLab is a very useful tool to analyse the detrimental effects of temperature variations. In this paper, we specifically use TempLab to analyse the impact of temperature (i) on processing performance, (ii) on a tree routing protocol, and (iii) on CSMA-based MAC protocols, deriving insights that would have not been revealed using existing testbed installations.

My contributions. I am the main author of this paper and was responsible for TempLab's design and implementation. I wrote the vast majority of the paper in collaboration and discussion with the co-authors, and carried out the experiments in the evaluation section. The model-based temperature profile section (Section IV-B3) has been mostly written by Marco Zuniga, whereas some experiments highlighting the use of TempLab for testing processing performance were carried out by James Brown and Utz Roedig (Section VI-A). I presented the paper at IPSN'14.

©2014 IEEE.
ISBN: 978-1-4799-3146-0.
DOI: 10.1109/IPSN.2014.6846744.

Link: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6846744.

2. Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line;

3. In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink.

TempLab: A Testbed Infrastructure to Study the Impact of Temperature on Wireless Sensor Networks

Carlo Alberto Boano[†], Marco Zúñiga[§], James Brown[¶], Utz Roedig[¶], Chamath Keppitiyagama[‡], and Kay Römer[†]

[†]Institute for Technical Informatics, Graz University of Technology, Graz, Austria
[§]Embedded Software Group, Delft University of Technology, Delft, The Netherlands
[¶]School of Computing and Communications, Lancaster University, Lancaster, United Kingdom
[‡]SICS Swedish ICT, Kista, Sweden

E-Mail: [†]{cboano, roemer}@tugraz.at, [§]m.zuniga@tudelft.nl, [¶]{j.brown, u.roedig}@lancaster.ac.uk, [‡]chamath@sics.se

***Abstract*—Temperature has a strong impact on the operations of all electrical and electronic components. In wireless sensor nodes, temperature variations can lead to loss of synchronization, degradation of the link quality, or early battery depletion, and can therefore affect key network metrics such as throughput, delay, and lifetime. Considering that most outdoor deployments are exposed to strong temperature variations across time and space, a deep understanding of how temperature affects network protocols is fundamental to comprehend flaws in their design and to improve their performance. Existing testbed infrastructures, however, do not allow to systematically study the impact of temperature on wireless sensor networks.**

In this paper we present TempLab, an extension for wireless sensor network testbeds that allows to control the on-board temperature of sensor nodes and to study the effects of temperature variations on the network performance in a precise and repeatable fashion. TempLab can accurately reproduce traces recorded in outdoor environments with fine granularity, while minimizing the hardware costs and configuration overhead. We use TempLab to analyse the detrimental effects of temperature variations (i) on processing performance, (ii) on a tree routing protocol, and (iii) on CSMA-based MAC protocols, deriving insights that would have not been revealed using existing testbed installations.

Keywords*—*CSMA; Protocol Performance; RPL; Temperature; TempLab; Testbed; Wireless Sensor Networks.

I. Introduction

Research and industrial deployments have shown that the operations of wireless sensor networks are largely affected by the on-board temperature of sensor nodes.

Temperature variations may significantly affect, among others, clock drift [1], [2], [3], battery capacity and discharge [4], as well as the quality of wireless links [5], [6]. Depending on the packaging and deployment location, the electronics of wireless sensor nodes may experience a substantial temperature variation. Sunshine may easily heat a packaged sensor node up to 70 degrees Celsius – especially if the packaging absorbs infra-red (IR) radiation [7], [8], [9], and long-term outdoor deployments have shown that the on-board temperature can vary by as much as 35°C in one hour and 56°C over the course of a day [10]. This variation is sufficient to cause a frequency offset of more than 100 ppm on the crystal oscillator frequency [11], which can affect the rendezvous process of synchronous duty-cycled MAC protocols. Such temperature change is also enough to reduce the received signal strength between two sensor nodes by more than 6 dB [10], which can change the packet reception rate (PRR) of the link from 100% to 0%. Hence, a deep analysis of how temperature affects the operation of sensor networks is necessary to inform the design of dependable applications prior to deployment.

To achieve this goal, the use of precise simulation tools would represent the most economical solution. However, analytical models that can accurately predict the impact of temperature on specific network protocols or radio transceivers are hard to obtain due to the complexity of the involved physical processes, and the ability of testing directly on real-hardware is hence highly desirable. To quantify accurately the effect of temperature on real hardware, it is important to isolate it from other environmental phenomena such as humidity and rain. Setting up a pilot deployment of a sensor network or using an existing outdoor testbed facility to evaluate the impact of temperature does not represent an optimal solution, since meteorological conditions cannot be controlled, making it impossible to ensure repeatability across several experiments. Furthermore, the temperature profiles that can be tested are highly specific to the deployment's location; the time of the year in which the experiment is carried out; and seasonal temperature variations (since deployments can last several months). What is needed to overcome these limitations is hence an indoor experimental facility that allows researchers and system designers to mimic the temperature variations normally found in outdoor deployments in a fast and simple way in order to obtain a precise understanding of the impact of temperature on networking protocols and on large-scale networks with high temperature gradients.

Traditional indoor testbed facilities used to evaluate protocols and applications under realistic conditions in a cost effective manner such as MoteLab [12], TWIST [13], Kansei [14], and NetEye [15], do not allow the evaluation of temperature effects. To date, a low-cost flexible testbed infrastructure that allows the repeatable generation of predefined temperature patterns across a sensor network still does not exist. Industry makes heavy use of temperature chambers during device verification processes (e.g., to calibrate sensors and transceivers [16]), but such solutions are not suitable due to their high cost and because they target individual components and not a network of nodes, which is necessary to disclose limitations at the communication level. *We aim to close this gap and design tools to make a testbed capable of reproducing real-world temperature profiles.*

Augmenting a testbed with the ability to reproduce temperature profiles is not a trivial task. Firstly, we need to recreate in a faithful manner the temperature variations that each node would experience in a real-world deployment over time. This can be achieved either by using temperature models tuned to a particular deployment site or by capturing temperature traces for extended periods of time. Secondly, these temperature profiles must be applied in such a way that no other property of the setup besides temperature is altered. Thirdly, the temperature profiles reproduced in the testbed need to be repeatable in order to allow a systematic quantification of the impact of temperature, and should emulate daily or seasonal changes within a few hours, allowing fast prototyping and experimentation. All these goals should be met while minimizing costs and efforts, so to have a widely applicable solution.

In this paper we present TempLab, an extension for wireless sensor networks testbeds that allows the on-board temperature of sensor nodes to be varied in a fine-grained and repeatable fashion. Our contributions are two-folded.

1. TempLab infrastructure. We describe testbed components, methods for implementing different temperature profiles, and evaluate TempLab to show that it can accurately reproduce temperature dynamics found in outdoor environments with fine granularity.

2. TempLab use cases. We use TempLab to examine and quantify the effects of temperature variations on sensornet applications and protocols, showing that they can drastically change the topology of a network and lead to network partitions, reduce significantly the performance of MAC protocols, as well as increase the processing delay in the network. These findings represent challenges to the sensornet research community and may open up a new research area of “temperature-awareness”.

This paper proceeds as follows. The next section motivates the need for a testbed solution to evaluate the impact of temperature on wireless sensor networks. Sect. III describes the requirements of such a testbed infrastructure. We describe the design and implementation of TempLab in Sect. IV, and investigate its performance in Sect. V, showing that temperature dynamics found in typical deployments can be accurately reproduced. Thereafter, in Sect. VI, we use TempLab to analyse the detrimental effects of temperature variations on sensornet applications and protocols. After describing related work in Sect. VII, we conclude our paper in Sect. VIII.

II. Temperature Matters

Temperature affects the operations of the most basic elements in *all* electric and electronic circuits: from resistors and capacitors to clocks and transistors. Due to this impact, assessing the effect of temperature on *individual* devices is usual practice in industry, and most electronic devices are given an operational range. Temperature also matters at the *network* level, but the effect of temperature on *inter-device* operation is far less understood.

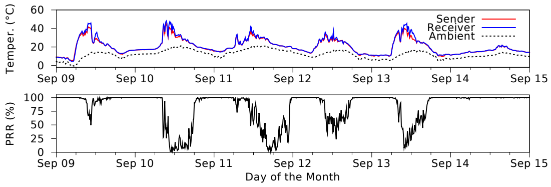


Fig. 1. Temperature has a strong impact on link quality in outdoor deployments. Even the normal temperature fluctuations during a day can render a good link useless [17].

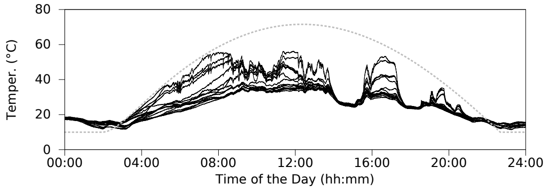


Fig. 2. Temperature profiles over the course of a day of 16 nodes deployed in an outdoor setting (solid curves), and maximum temperature profile obtained with the model presented in Sect. IV-B3 (dashed curve).

A few studies have started to evaluate the effect of temperature on network operations. Bannister et al. [5] showed that temperature has a significant impact on link quality, and in our earlier work, we validated these claims with a more systematic study, showing that temperature affects in a similar way different hardware platforms [10]. The most powerful case highlighting the importance of temperature at the network level is probably given by Wennerström et al. [17], who report insights from a long-term study showcasing the impact of meteorological conditions on the quality of 802.15.4 links. Fig. 1, based on traces recorded during Wennerström’s outdoor deployment, shows the on-board temperature of a transmitter and receiver pair, and the packet reception rate of their link: even the normal temperature fluctuations occurring during a day can transform a perfect link (100% PRR) into an almost useless one ($\approx 0\%$ PRR).

However, besides these initial studies, temperature has not received (at the network level) the same level of attention that it received at the device level, but it definitely should. Temperature introduces a sort of *dynamic heterogeneity* across the network: two nodes with the same parameters, but with different on-board temperatures, will perform differently. It is important to analyse this temperature-based heterogeneity, because even nodes that are physically close can have vastly different temperature profiles, as already reported by several real-world deployments [3], [18], [19].

In Wennerström’s deployment [17], for example, all the nodes are within each-other’s transmission range, and experience highly different temperatures. Fig. 2 depicts the on-board temperature of sixteen of these nodes over the course of a summer day [17], and Fig. 3 depicts the temperature density function for two of them. One node is much “hotter” than the other, and this hot node will have a shorter transmission coverage [5], [10], a larger clock drift [2], whereas the lifetime of the cold node will be much shorter [4].

How do all these temperature effects, and others that are yet uncovered, affect the operation of network protocols?

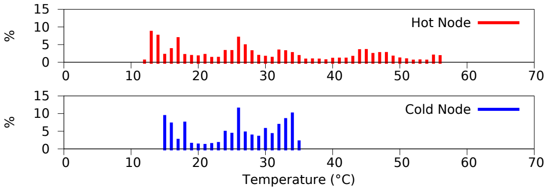


Fig. 3. The temperature profile of nodes can be highly different even if nodes are in proximity of each other. This difference can affect the overall performance of the network.

To evaluate these effects, we need to provide the sensor network community with a simple, yet accurate, low-cost testbed infrastructure enabling the study of the effects of temperature variations on the network performance in a precise and repeatable fashion.

III. Requirements

Such a testbed solution should essentially have the ability to control the on-board temperature of wireless sensor nodes. However, in order to accurately reproduce the temperature dynamics that can be found in typical deployments, it is not simply enough to choose off-the-shelf heating and cooling elements and connect them to the testbed. The choice of the hardware, as well as the design of the infrastructure should meet a number of requirements that we describe below.

Large temperature range. Ideally, the testbed would be able to reproduce temperature patterns covering the complete operating range of sensor nodes. For example, in the case of the off-the-shelf TelosB platform, this would imply to heat sensor nodes up to 85°C, but also to cool them down to -45°C, according to the datasheet. While it is perhaps not necessary to reach the limits of the operating range, it is important to reproduce the conditions that can be found in a real deployment during the hottest and coldest times of the year. In particular, the testbed should be able to reach values up to 70-75°C, as experiences from outdoor deployments have shown that the on-board temperature of sensor nodes that are exposed to direct sunlight or are embedded into transparent packaging can reach extremely high values [7], [8], [9], [18]. Similarly, it is important to reach temperatures below 0°C to reproduce the conditions that can be found in a real deployment during the coldest times of the year.

Fine-grained temperature control. As shown in Fig. 2, the temperature of a node deployed outdoors can continuously vary depending on the presence of sunshine and obstacles (e.g., clouds or buildings). These effects cause *continuum* gradients of temperature, i.e., the jumps of temperature are not sudden and discrete, but smooth. Since our goal is to recreate temperature traces in the most faithful manner, the testbed infrastructure should be able to precisely tune the on-board temperature of a sensor node with a high resolution.

Fast temperature variations. In a real deployment, temperature can change quickly due to meteorological effects such as wind, rain, and snow, as well as due to the presence of clouds or sunshine. In the deployment shown in Fig. 2, for example, a node that receives the first sun-rays at the beginning of the day increases its temperature as much as 1.98°C/minute. An important requirement for the infrastructure that we want to build is hence the ability of reproducing these variations as fast at they occur in the real-world. This requirement has a strong effect on how accurately temperature dynamics can be reproduced.

Time scaling. It is often desirable to compress the time scale of an experiment to save evaluation time (as long as such time compression does not depend on the rate of the temperature change, but only on the absolute temperature values). One may want to time-lapse the recreation of real-world traces and playback, for instance, in a few hours the profile of a full day. This poses stronger requirements on the ability of the testbed to quickly heat up and cool down nodes.

Per-node temperature control. As observed in Fig. 2, the profile of each node can be highly different. Hence, placing all the nodes into a single chamber would not be realistic because all nodes would follow the same temperature profile. Temperature must be controlled individually on each node.

Unaltered system behaviour. The extension of the existing infrastructure should ideally not alter the behaviour of the system in any way, as this may lead to unwanted (and unexpected) system failures. For example, the use of metal casings should be restrained, as RF propagation should be minimally affected. Similarly, the use of I/O ports of a sensor node to control heating or cooling devices has to be avoided if this would affect the operations of the system.

Scalability. Although it may not be necessary to augment all nodes of an existing infrastructure with temperature control, it should be ideally possible to extend an entire testbed. Commonly used testbeds such as MoteLab [12], TWIST [13], and NetEye [15] have typically up to 200 nodes, and our testbed solution should be able to scale at these levels.

Low cost. All the above requirements have to be satisfied while minimizing the cost of the solution, in order to make it applicable on a large-scale.

IV. TempLab: Architecture and Implementation

In this section, we present the general architecture of TempLab, our low-cost extension of testbed facilities capable of reproducing real-world temperature profiles with fine granularity, and describe the hardware and software components that we use in our implementation.

A. Architecture

In order to study the effects of temperature variations on the operation of wireless sensor networks and their protocols, the infrastructure needs to be able to reproduce specific temperature profiles on several nodes. This requires (i) temperature profiles to be reproduced, (ii) actuators to control the on-board temperature of each sensor node, and (iii) a controller that uses the actuators to instantiate the desired profiles.

1) Temperature Profiles: In order to support a wide range of experimentation techniques, TempLab can generate temperature profiles using three different approaches.

Firstly, one can re-play temperature traces collected in-situ at a given deployment site, such as those in Fig. 2. Such *trace-based* temperature profile instantiation can accurately reflect

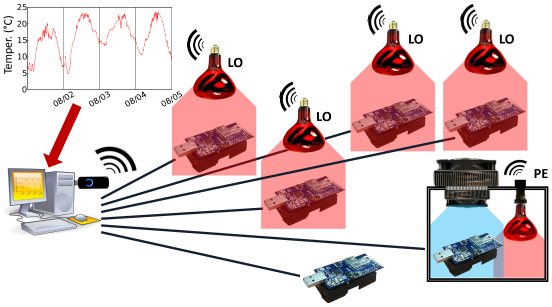


Fig. 4. Sketch of TempLab's architecture.

the temperature variations over time with fine granularity if long-term measurements from one or more nodes are available. However, traces are not always at one's disposal, or they may be incomplete: trace-based profiles can be used only if one or more sensor nodes deployed previously actually collected temperature data with a frequency sufficiently high to capture the dynamics of temperature changes.

A second possibility is, therefore, to use a *model-based* temperature profile to have an estimation about the temperature dynamics at a certain location without the need of traces collected in-situ. A model-based approach uses models to estimate the temperature profile of objects using basic environmental information such as the maximum solar radiation and the minimum temperature during a day (that is readily available from satellites and meteorological stations). We derive such a model in Sect. IV-B3.

A third possibility is to use TempLab to vary the temperature of sensor nodes using specific *test patterns*. For example, a user may not be interested in recreating a specific profile and needs instead only to verify whether a high temperature variation has an impact on the operation of a given protocol. In this case, TempLab can be fed with on-off patterns (e.g., a series of cold and warm periods) or jig-saw patterns that vary temperature with a specified frequency, allowing a quick debugging of protocols' behaviour.

2) Actuators: To heat-up and cool-down the on-board temperature of sensor nodes, one or more actuators are required for each node. Actuation can be applied *out-of-band* or *in-band*. Out-of-band means that the sensor node is not involved in the control of its temperature, i.e., additional processing hardware is needed. In-band methods, instead, make use of the sensor node to vary its on-board temperature, e.g., by using its I/O pins to control heating or cooling devices. Although in-band methods have the advantage of avoiding extra-hardware (and reduce testbed costs), they may alter the system behaviour and violate the corresponding requirement.

Therefore, we design TempLab following an *out-of-band* approach based on infra-red heating lamps and cooling enclosures that allow to vary the on-board temperature of wireless sensor nodes in the range [-5, +80] °C. TempLab can have two types of nodes with different capabilities as shown in Fig. 4: *LO* and *PE* nodes. *LO nodes*, which stands for lamps-only nodes, are heating-only devices that have the capability of warming the sensor nodes between room temperature and their maximum operating range. They are based on IR heating lamps supported by Polystyrene hard foam, and they do not have any capability to cool-down the nodes below room temperature. *PE nodes*, which stands for Peltier enclosure nodes, are hard temperature-isolating Polystyrene enclosures with an embedded IR heating lamp and an air-to-air Peltier module to heat-up and cool-down the inner temperature of the casing[1]. To control the intensity of the IR lamps and the operations of the Peltier module, we borrow existing home automation solutions and use *wireless dimmers* to vary the intensity of the lamps and *on-off wireless switches* to control the Peltier modules embedded in the enclosure. To make sure that the temperature control system does not interfere with the existing testbed communication, we select home automation solutions working on a ISM frequency band that is different from the one used by the sensor nodes.

This approach can easily scale to large testbeds as PE and LO nodes only need to be plugged into wall outlets and require no further cabling. Furthermore, home automation solutions such as Z-Wave allow multiple devices (LO and PE lamps in our case) to communicate in a multi-hop fashion, and create different home networks each of which can have a maximum of 256 nodes. Each Z-Wave module can act as an RF repeater and commands can be routed through a maximum of four devices. This gives each home network a maximum range of 122 meters and routing is managed automatically [21], and can hence in principle scale to large buildings. If a very large number of nodes need to be supported, it is possible to partition the control network and use several controllers. All what is needed is the availability of a power line, but as in most indoor testbeds there exists a wired back-link to each node, the efforts to add a power line are typically not too high.

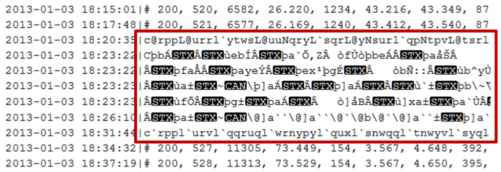
3) Controller: To instantiate a temperature profile and control heat lamps and Peltier modules, TempLab uses different controllers running on a centralized testbed gateway computer.

Open-loop controller. The simplest one is an open-loop controller that varies the intensity of the light bulbs in LO and PE nodes according to a pre-computed calibration function[2]. This is possible if the impact of each dimming level on the on-board temperature of a node is known based on a previous calibration. In this case, the open-loop controller can instantiate a given profile without further processing. The key advantage of this approach is hence that no sensors are needed to measure the actual temperature of the motes during the experiment. For an accurate replay of temperature dynamics, however, the surrounding environment as found during calibration would need to remain constant, as the controller would not account for external factors influencing temperature such as open windows or sun shining in the room hosting the testbed.

Closed-loop controller. To precisely regenerate trace- or model-based temperature profiles, TempLab uses a closed-loop proportional-integral (PI) controller that tries to minimize the difference between the desired temperature profile and the on-board temperature of the sensor node of interest. The controller

[1] We have selected polystyrene-based materials to have a low impact on signal strength, as the RF absorption is minimal [20].

[2] For PE nodes, one can vary the intensity of the heat lamps while the Peltier module is constantly active. As we show in Sect. V, the IR lamp can change the temperature much quicker than the Peltier module, and a constantly active Peltier module does not slow down the heating from the IR lamp significantly.



```
2013-01-03 18:15:01|# 200, 520, 6582, 26.220, 1234, 43.216, 43.349, 87
2013-01-03 18:17:48|# 200, 521, 6577, 26.169, 1240, 43.412, 43.540, 87
2013-01-03 18:20:35|c@rppL@urrl`ytwsL@uuNqryL`sqrL@yNsurl`qpNtpvL@tsrl
2013-01-03 18:23:22|CþbÁSTXÂSTXùebÍÂSTXþa`Õ,ZÂ òfÙòþbeÁÂSTXþaåŠÂ
2013-01-03 18:23:22|ÂSTXþfaÅÂSTXþayeÝÂSTXþex¹þgÉSTXÂ   òbÑ::ÂSTXùb^yÙ
2013-01-03 18:23:23|ÂSTXùa±STX~CAN\þ]aÁSTXÂSTXþ]aÁSTXÂSTXù`±STXþb\~
2013-01-03 18:23:23|ÂSTXùfÕÂSTXþg±STXþaÁSTXÂ   ò]åBÂSTXù]xa±STXþa`ÙÂ
2013-01-03 18:26:10|ÂSTXþa±STX~CAN\@]a``\@]a``\@`\@b\@`\@]a``±STXþ]a`
2013-01-03 18:31:44|c`rppl`urvl`qqruql`wrnypyl`quxl`snwqql`tnwyvl`syql
2013-01-03 18:34:32|# 200, 527, 11305, 73.449, 154, 3.567, 4.648, 392,
2013-01-03 18:37:19|# 200, 528, 11313, 73.529, 154, 3.567, 4.650, 395,
```

Fig. 5. Unreadable serial output in the presence of sudden thermal variations.

should hence receive a periodic feedback with frequency F_U about the on-board temperature of the sensor node in order to minimize the error with respect to the desired temperature profile. The reading of the on-board node temperatures can be carried out either *out-of-band* through the use of an external device or *in-band* using the sensor node itself to measure the temperature and forward it to the controller. As most off-the-shelf wireless sensor nodes carry on-board a temperature sensor, it is very tempting to use an in-band approach to provide up-to-date temperature measurements without adding extra-costs. However, it has to be ensured that system behaviour is not altered. TempLab uses an in-band approach using the USB back-channel to periodically convey temperature readings to the controller. This task is carried out using a low-priority routine executing only when the processor is idle.

During our experiments, we have observed that common USB serial connections used in testbeds for data logging and node programming may be unable to cope with very fast temperature fluctuations, as they result in de-synchronization of the USB sender and receiver. In the presence of such variations, the USB serial port looses synchronization with the mote and the characters forwarded to the USB back-channel become temporarily unreadable, as shown in Fig. 5. Since standard nodes do not handle this issue autonomously, TempLab either re-initializes the USB port or piggybacks the temperature readings onto regular data packets. In this way, other nodes that do not suffer from this issue can report the temperature to the controller over the USB back-channel.

B. Implementation

We now describe the hardware and software components that we used to extend our local university testbed based on Maxfor MTM-CM5000MSP nodes (TelosB replicas).

1) Hardware: In our implementation, we use Philips E27 IR 100W light bulbs that can be remotely dimmed using the Z-Wave wireless home automation standard. The latter operates on the 868 MHz ISM band, and hence does not interfere with the communications between the wireless sensor nodes (that use the 2.4 GHz ISM band)[3]. To vary the intensity of the light bulbs, we used Vesternet EVR_AD1422 Z-Wave Everspring wireless dimmers, which provide 100 dimming levels.

LO nodes are only controllable using dimmers. PE nodes have the capability of going below room temperature thanks to 4 cm-thick enclosures made of hard Polystyrene foam embedding, in addition to the IR heating bulb, an ATA-050-24 Peltier air-to-air assembly module by Custom Thermoelectric [22]. The latter allows on-board temperatures of -5°C when operated at room temperature (about 20°C), and can be controlled through Vesternet EVR_AN1572 Z-Wave Everspring on-off wireless switches. The Polystyrene hard foam isolating box has a minimal impact to the radio propagation of sensor nodes and supports temperatures up to +85°C. The overall hardware cost is €65 for each LO node, and €293 for a PE node[4].

2) Software: We now provide further details on the software used to develop TempLab.

Actuators. We control the Z-Wave network with a C++ program that uses the Open Z-Wave stack to vary the intensity of dimmers and duty cycle the Peltier modules. Commands to the control network are sent through the Aeon Labs Series 2 USB Controller deployed within the testbed facility.

Sensor nodes. Each node runs Contiki, and contains a low-priority process that periodically measures temperature using the on-board SHT11 sensor, and communicates the readings over the USB back-channel. This can be also easily implemented in TinyOS or other operating systems, since it needs only basic building blocks such as reading and outputting temperature. To select the sampling frequency F_U, i.e., how often should the controller receive feedback about the on-board node temperature and update the intensity of the IR lamps, we use the fastest temperature variation observed in the outdoor deployment shown in Fig. 2, and compare it to the accuracy of the on-board temperature sensors. In our case, the nodes carry SHT11 sensors that have an accuracy of 0.4°C. According to the profiles shown in Fig. 2, such a variation can be reached within 12 seconds.

Controller. The PI controller is implemented as a standalone multi-threaded C++ application executing on the testbed gateway that receives as input a file with two columns: the first one contains the time of the day, the second one describes the on-board temperature that the node should have at that time. The controller is agnostic to the type of trace (whether derived empirically or from a model): as long as the file adheres to the two column format, it will (try to) recreate such temperatures based on this information and the feedback signals from the motes. In case the user chooses to time-lapse the experiment, the controller skips rows accordingly, e.g., for a 2x speed, the controller omits every other line. Users can manually assign the available traces to the temperature-controlled nodes in the network, and if a non-implementable mapping is created, the controller will signal an error. The parameters P and I of the controller have been found empirically by testing the response of the system with extreme temperature variations and by choosing those values that achieve high stability and minimal overshooting. We found that P=2 and I=0.01 is the empirically best configuration of the controller that achieves fast and self-stabilizing control.

3) Deriving Model-based Temperature Profiles: Using thermodynamic equations we now derive a temperature model suitable to create temperature profiles for nodes. We focus on outdoor deployments where IR radiation from the sun and air temperature are the most significant factors.

[3] We have also implemented a TempLab version that uses the LightwaveRF standard operating on the 433 MHz ISM band, in case the sensor nodes in the testbed operate on the 868 MHz ISM band. In the rest of the paper we refer to the Z-Wave implementation.

[4] The high price of the PE nodes is due to the fact that we have chosen off-the-shelf Peltier assembly modules [22]. In principle, one could simply buy the individual components (e.g., a fan or a Peltier element) to speed up the cooling of LO nodes at a much lower price.

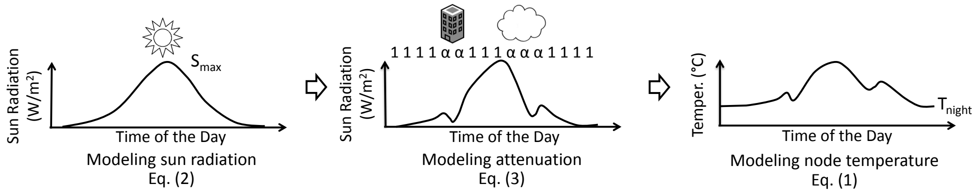


Fig. 6. Model-based temperature profile generation.

Energy absorption and dissipation. In essence, objects heat up by absorbing solar radiation and cool down by constantly releasing energy to their surrounding: the balance between these processes determines the object temperature. An object that is exposed to the sun, absorbs energy according to: $E_{in} = S\alpha A\Delta t$, where S is the solar radiation, α is the attenuation of the solar radiation, A is the exposed area of the object and Δt is the amount of time exposed to the solar radiation. On the other hand, objects release energy according to: $E_{out} = sT^4 A\Delta t$, where s is the Boltzmann constant and T is the temperature of the object in Kelvin.

Energy balance. Considering the energy absorption and energy dissipation of an object, its change of temperature ΔT is determined by the heat energy equation: $H = C_p m\Delta T = E_{in} - E_{out}$, where C_p is the specific heat of the object and m its mass. The temperature of an object cannot be less than air temperature at any given time t (T_t^{air}). Hence, at time $t + \Delta t$, the object temperature is given by:

$$T_{t+\Delta t} = \max\{\ T_t + \frac{(S_t\alpha_t - sT^4)}{C_p m} A\Delta t,\ \ T_t^{air}\ \} \quad (1)$$

Considering a standard mote with parameters m = 50 grams, C_p = 0.5 $\frac{J}{gC}$, A = 20 cm^2; the model only requires the sun radiation S_t, the air temperature T_t^{air} and the attenuation α_t ($0 \leq \alpha_t \leq 1$).

Sun radiation and cloud obstruction. In the absence of any obstructions, the sun radiation throughout the day can be modelled by a gaussian-like shape [23]:

$$\begin{aligned} S_t &= \frac{S_{max}}{\max\{\mathcal{N}(0,\sigma)\}} \frac{1}{\sqrt{2\pi}\sigma} \exp^{-(t-\delta)^2/2\sigma^2} \\ &= S_{max} \exp^{-(t-\delta)^2/2\sigma^2}, 0 \leq t \leq 2\delta \end{aligned} \quad (2)$$

where S_{max} is the maximum sun radiation during the day, and $t = 0$ and $t = 2\delta$ represent the 00 hrs and the 24 hrs. The number of hours with sun light (length of day) can be fine-tuned with σ and δ. To further simplify Eq. 1, instead of considering the air temperature throughout the day (T_t^{air}), we use only the minimum temperature in the day (night temperature T_{min}). Hence, at this point, the only information that we need to model the *clear sky* temperature of a node is the maximum radiation and minimum *air* temperature.

Few locations, however, receive constant sun radiation throughout the day. In most scenarios, clouds block the sun radiation and cause sudden variations of temperature. The length of clouds and the length of the clear sky between clouds are known to have exponential distributions $\lambda \exp^{-\lambda x}, x \geq 0$, with $\frac{1}{\lambda}$ representing the average cloud (or inter-cloud) length [24]. Denoting $\overrightarrow{\alpha}$ as an *attenuation vector* where all elements are α and its length is given by the exponentially random length of a cloud. And denoting $\overrightarrow{1}$ as a *clear-sky vector* where all elements are 1 (i.e. $\alpha = 1$) and its length is equal to the random length of an inter-cloud period; the variable α_t in Eq. 1 is the t^{th} element of the vector:

$$\overrightarrow{v} = \{\overrightarrow{\alpha_1}, \overrightarrow{1_1}, \ldots, \overrightarrow{\alpha_i}, \overrightarrow{1_i}, \ldots\}. \quad (3)$$

At each t in Eq. 1, the t^{th} element is used to capture the amount of sun radiation attenuated during the respective period Δt. The shade of events that are specific to the scenario of interest (trees, buildings, etc), can be included in $\overrightarrow{v}$ by inserting attenuation elements (α) in the vector.

The model can be easily coded using any programming or scripting language. In TempLab, we use Matlab, making sure that the output of the model adheres to the requirements of the PI controller. To compute a temperature value at time t, Eq. 1 is evaluated for the respective value of Δt. Fig. 6 captures the steps followed by the model, and the outcome is a curve similar to the dotted one shown in Fig. 2, or one with random fluctuations due to shades.

The model allows the user to test a wide range of scenarios. The user can test the worst-case temperature with clear skies, generate shades of any length at any time (to test temperature gradients), and generate random instances for each node by varying the model parameters.

V. Evaluation

In this section, we carry out an experimental evaluation of the capabilities of our TempLab implementation. First, we investigate the performance of TempLab in terms of implementable temperature profile dynamics and highlight the limitations on how fast nodes can be heated or cooled. Thereafter, we show that temperature dynamics found in typical deployments can be accurately reproduced despite the low-cost infrastructure, even when compressing the time scale of an experiment to save evaluation time.

A. Heating and Cooling Limits

To verify how fast LO and PE nodes can be heated and cooled, we carry out an experiment in which we let the closed-loop PI controller heat the nodes to 80°C with an update frequency F_U of 2 seconds. The initial temperature is room temperature for LO nodes and 0°C for PE nodes, respectively. After reaching a stable temperature, the controller cools the nodes down to their original value.

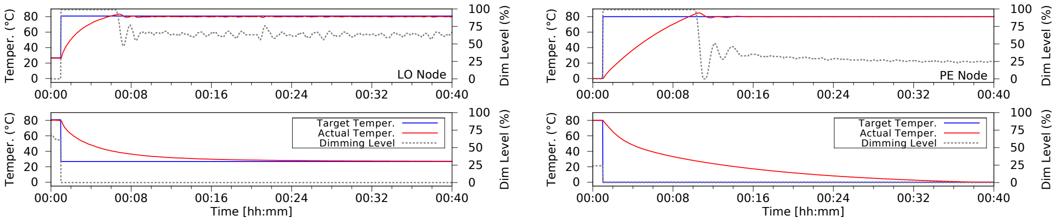


Fig. 7. Limits in the speed of heating and cooling for LO and PE nodes.

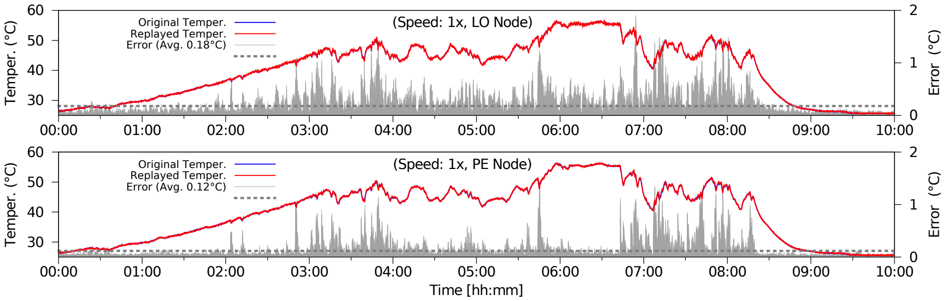


Fig. 8. Accuracy of LO and PE nodes in replaying a real-world trace captured during summer.

1) LO nodes: Fig. 7 (left) shows that LO nodes can heat from room temperature (26°C) to 80°C in less than 5 minutes, with an average heating slope of 11.3°C/minute. As LO nodes do not have cooling capabilities, their cooling is rather slow: they need only 7 minutes to decrease from 80°C to 35°C, but they require the same time to decrease from 35°C to 30°C, and 20 more minutes to get back to 26°C.

2) PE nodes: Fig. 7 (right) shows that PE nodes can heat from 0°C to 80°C in less than 9 minutes, with an average heating slope of 9.3°C/minute. PE nodes are obviously much more efficient in cooling than LO nodes: they need only 6 minutes to decrease from 80°C to 35°C, and 10 minutes to decrease to ambient temperature (26°C). Overall, they can vary the temperature from 80°C to 0°C in less than 35 minutes.

B. Regeneration of Traces

We now evaluate TempLab's ability of reproducing a given temperature profile. We compute the accuracy of TempLab by computing how close the instantiated temperature profile P_I follows the given profile to be reproduced P_G. The overall accuracy Q_n of the reproduced temperature profile at node n can be expressed as:

$$Q_n = \frac{1}{T}\int_0^T |\mathrm{P_I(t) - P(t)}|\,\mathrm{d}t \qquad (4)$$

where T is the duration of the experiment. Besides the requirement to follow a temperature profile over time, it is also important to ensure that the rate of temperature changes is reflected accurately. At no point in time the instantiated temperature curve at a node n should deviate too much from the given temperature profile. The maximum deviation q_n can be expressed as:

$$q_n = \max_t |P_I(t) - P(t)| \qquad (5)$$

The smaller the value of Q_n, the better the instantiation of the temperature profile, whereas the smaller q_n, the better the dynamics of the temperature change are reflected.

We take as a reference for our evaluation two temperature traces collected in an outdoor deployment in Sweden [17]: one taken during summer (August), and a "colder" one taken in the end of October, when temperature approaches 0°C.

1) Summer trace: Fig. 8 shows that both LO and PE nodes can instantiate the desired temperature profile on the sensor nodes with very high accuracy. The average error Q_n equals 0.18°C and 0.12°C, whereas q_n is 1.90°C and 1.43°C for LO and PE nodes, respectively. This is a remarkable accuracy, and shows that despite the use of low-cost components (LO nodes), TempLab can still reproduce with high accuracy real-world temperature profiles above room temperature.

2) Winter trace: During winter time, the sun can quickly heat up the temperature in the package hosting the sensor nodes. We replay a trace captured during October 2012 [17], in which the on-board temperature of a node has a significant variation from 45°C during daytime to 0°C in the evening, and see how accurately PE nodes can instantiate this temperature profile on sensor nodes. Fig. 9 shows the results: the average error Q_n equals 0.14°C, whereas $q_n = 3.36$°C[5].

3) Accuracy of time-lapsed traces: The accuracy of the replay shown in Fig. 9 is even more remarkable if we consider that we have compressed the original 24-hour trace into 8 hours playback time, i.e., we used a compression factor of 3. We now show the accuracy of LO and PE nodes in the regeneration of traces in which the time has been compressed even further.

[5] It is important to highlight that LO nodes have a lower granularity than PE nodes. Our experimental results show that this does not affect the accuracy of the system in a distinguishable manner, as the controller does a commendable job in compensating such differences.

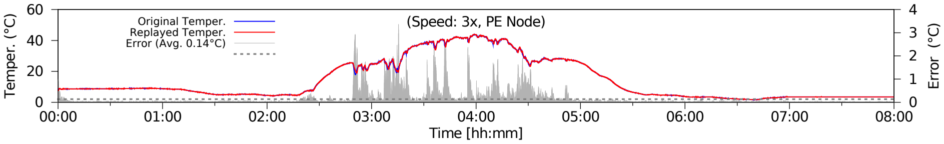


Fig. 9. Accuracy of PE nodes in replaying a real-world trace captured during winter.

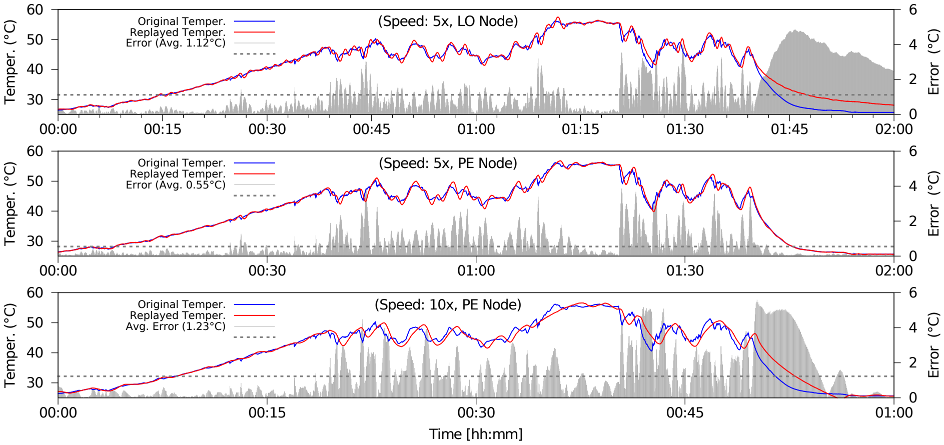


Fig. 10. Accuracy of LO and PE nodes when compressing the time-scale of the experiment.

Fig. 10 shows the results: when instantiating the same trace used in Fig. 8, LO nodes show evident limits due to the lack of cooling capabilities. Compared to the error of 0.18°C when regenerating at normal speed, the average error Q_n raises to 1.12°C when the time is compressed by a factor of 5, whereas Q_n is 0.52°C and 1.90°C when replaying a trace compressed with factor 3 and 10, respectively.

PE nodes, instead, can replay a trace 5 times faster than the original speed with $Q_n = 0.55$°C (the error is halved compared to the LO nodes) and $q_n = 3.84$°C. When compressing time by a factor of 10, however, we can start to observe that the Peltier modules reach their limit, and cannot properly cool down in only 4 minutes what in reality takes 45 minutes. Nevertheless, Q_n is only 1.23°C, and q_n = 5.57°C.

VI. TempLab in Action

In this section we present a series of experiments carried out using TempLab. We demonstrate that temperature has a significant impact on processing and protocol performance, and show that TempLab is an ideal tool to investigate these effects. Our aim is not to give a complete solution to the issues that we reveal, but rather to highlight to the community several research challenges that require attention. We believe that TempLab can play a significant role in this emerging research area.

A. *Testing Processing Performance*

Many sensornet applications require a significant amount of on-node processing, so that data is filtered, analysed, or aggregated before being delivered over the network. Heavy processing is often also required for compression, i.e., to reduce the volume of data that has to be transmitted. The processing time required to compress, filter, or analyse data is very significant, as it defines the achievable sampling rate and determines if deadlines can be fulfilled. In [25], the execution of an object detection algorithm requires 240 ms for an image size of 128x128 pixel on an ATmega128 running at 7.3728 MHz. In structural health monitoring application such as [26], accelerometer samples are compressed before transmission, which requires 17.32 ms on a platform employing an MSP430 running at 4 MHz. In medical applications such as ECG monitoring, depending on the algorithm used, the compression of two seconds of ECG data (512 samples) can require up to 580ms [27]. Similarly, applications that require encryption algorithms also require significant processing: software based encryption and authentication of a packet with 56 byte payload requires 17.3 ms on a TelosB platform employing an MSP430 running at 4 MHz [28].

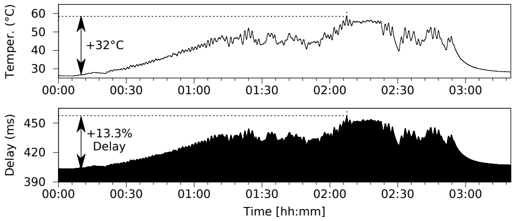


Fig. 11. Inter-arrival times follow temperature variations.

We will now show that temperature can have a significant impact on the processing capabilities of a node, and that these execution times may significantly vary when the processing node is deployed outdoors. We use TempLab to mimic the operations of the class of applications previously discussed. We develop a Contiki application in which data is processed using a fixed-effort of 1 million processor cycles and the result is transmitted to a sink node. To this end, we use Maxfor MTM-CM5000MSP nodes employing *nullrdc*, a simple MAC protocol without duty-cycling. This makes sure that we avoid protocol-specific effects.

First, we run the application in a testbed without any temperature variation, i.e., we leave the nodes at room temperature. The inter-arrival time of messages at the sink is, on average, 404.35 ms with a tiny variation of 0.88 ms. Next we assume that the application will be used outdoors, and we expose the processing node to temperature variations. In particular, we use the same summer trace profile used in Sect. V-B to mimic a temperature profile to which a node would be exposed during summer. Fig. 11 shows the obtained inter-arrival times when the processing node is cycled through a time-lapsed version of the 10-hour trace. At the (lowest) temperature of 26°C the inter-arrival time is 402.9 ms, whilst at the highest temperature of 58°C an inter-arrival time of 456.5 ms is observed. This represents a change of 13.3% for an increase of 32°C, hence the variation in temperature introduced a significant change.

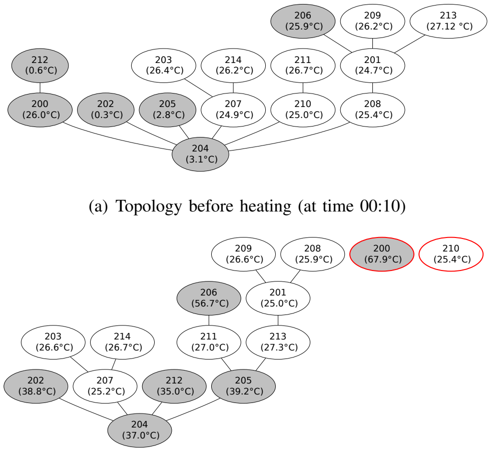
Closer investigation reveals that processing requires significantly more time on hot nodes than on cold ones. We use TempLab to test a simple application toggling a GPIO pin after a fixed amount of processor cycles, and recording the time required to complete this amount of work. The anomalous behaviour is indeed caused by the temperature-dependent drift of the processor clock: when temperature is increased from 21 to 54°C, the processor speed drops by roughly 13%.

Although this outcome is not really surprising, it would not have been possible to verify that the application performance would be largely affected in the expected target area (and assess by how much) when using a standard testbed without temperature control. Using TempLab, the analysis of sensornet performance under varying temperature becomes very simple and helps to identify crucial performance aspects. Although in this paper we do not discuss a solution, TempLab can also be used also to find a solution to the problem and to evaluate its effectiveness, e.g., a periodic recalibration of the processor clock with the temperature stable external crystal.

B. Testing Protocol Performance

In this section, we use TempLab to highlight the strong impact of temperature on wireless communication, routing topologies, and MAC protocols. Especially relevant when analysing protocol performance is TempLab's ability to generate specific test patterns, as well as the possibility to heat separately and/or simultaneously transmitter and receiver nodes, which is fundamental to systematically study the impact of temperature on different protocol components [10]. Carrying out similar experiments using multiple thermal chambers would be infeasible, due to the high costs of single units, and to the implications on the propagation of signals due to the metal casing.

1) Impact of Temperature on Routing Protocols: Earlier work has shown that temperature affects the efficiency of low-power wireless radios and hence the quality of links [5]. However, an experimental evaluation of how temperature variations affect network protocols is, to date, still missing. We now use TempLab to show how temperature fluctuations can affect the behaviour of the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [29].



(a) Topology before heating (at time 00:10)

(b) Topology after heating (at time 01:00)

Fig. 12. An increase in temperature can lead to drastic changes in the RPL topology, including a network partition and an increase in network diameter.

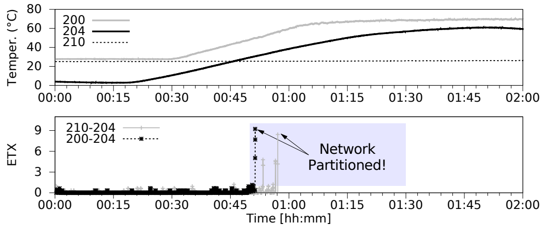
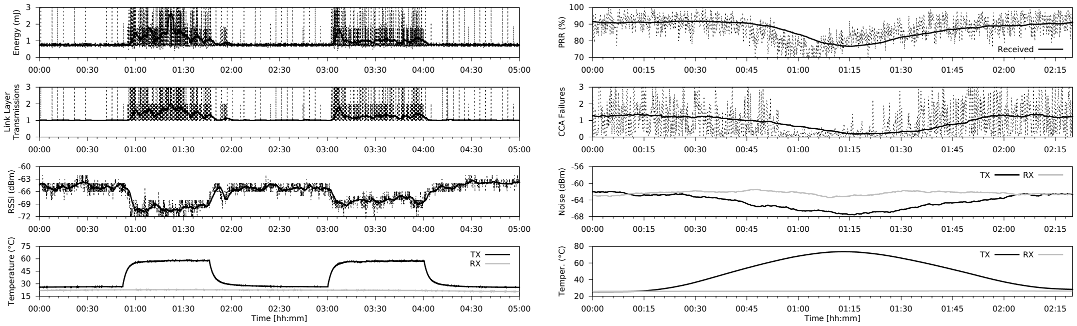


Fig. 13. Sudden rise of ETX while temperature increases.

We program fifteen Maxfor MTM-CM5000MSP nodes in our local testbed with a basic Contiki application that uses ContikiRPL [30]. Each node sends a message to the root node (id 204) every minute and logs the transmitted and received packets, as well as the on-board temperature and the Expected Transmission Count (ETX) of the active links. We use TempLab to evaluate the impact that daily fluctuations of temperature can have on the RPL topology with a test pattern that gradually increases the temperature of the designated root node and of one third of the other testbed nodes.

Fig. 12(a) shows a snapshot of the RPL topology at the beginning of the experiment, when nodes are kept at low temperature: all nodes are connected to the sink within a maximum of three hops. Fig. 12(b) illustrates a snapshot of the RPL topology after temperature has increased: temperature-controlled nodes are shaded in gray. *The increase in temperature led to drastic changes in the topology of the network, including a network partition and an increase in network diameter*. Nodes 200 and 210 had a direct link to the root node when temperature was low (Fig. 12(a)), but these links are isolated from the network once temperature has increased.



(a) Impact on MAC efficiency

(b) Impact on CCA efficiency

Fig. 14. Impact of temperature on CSMA-based MAC protocols: the energy expenditure increases as well as the amount of link-layer transmissions at high temperatures, due to a reduced efficiency of CCA leading to a higher packet loss rate.

As we have highlighted in Sect. IV-B3, the presence of direct sunshine on nodes or clouds can quickly vary the on-board temperature of sensor nodes. ContikiRPL attempts to construct a tree by minimizing the ETX sum along the paths to the root. However, ETX changes abruptly with fast temperature changes, especially when packets are exchanged sporadically. In our experiments, we can indeed observe a sudden increase of ETX in links 200 → 204 and 210 → 204 (Fig. 13), which will lead to a sudden network partition.

These results emphasize the need for techniques that infer the information about the on-board temperature of sensor nodes to the routing layer, so that the most stable tree can be computed before drastic temperature changes occur, as we have shown in [31]. Because of the stochastic nature of the topology formation on RPL (it depends on the *trickle timers* used on nodes to announce DAG information object messages), it is very important to test protocols against several temperature profiles, and TempLab can be a very handy tool to conveniently control and repeat temperature patterns.

2) Impact of Temperature on MAC Protocols: Temperature variations can also drastically affect the performance of medium access control protocols. It is not difficult to envision that protocols relying on tight time synchronization, such as the ones based on time-division multiple access (TDMA) schemes [32] [33], can be vulnerable to sudden temperature variations across the network due to clock drifts and slowdown of micro-controllers. In the context of TDMA protocols, TempLab can be used to experimentally find the optimal value for critical parameters such as slot size, guard time, and re-synchronization frequency, so that protocols can operate reliably despite challenging temperature variations that can occur at the final deployment site.

Less obvious is the fact that also the performance of carrier-sense multiple access (CSMA) protocols degrades because of temperature variations. In this section, we use TempLab to provide experimental evidence that *CSMA protocols may reduce their efficiency when operating at high temperatures*. We carry out experiments consisting of several transmitter-receiver pairs of Maxfor MTM-CM5000MSP nodes running a basic Contiki application, in which the transmitter node periodically sends packets to its intended receiver and collects statistics such as the overall energy expenditure at the link-layer and the noise in the radio channel. We compute the noise as the maximum of 20 consecutive cc2420_rssi() readings after the transmission of a packet. Receivers acknowledge the message reception and measure the RSSI of received packets as well as the noise in the channel after packet reception. We select ContikiMAC, Contiki's default MAC protocol, and use TempLab with a test pattern that progressively heats the transmitter while keeping the receiver at constant temperature.

Fig. 14(a) (top) shows that the overall energy spent at the link layer to successfully transmit a packet increases at high temperatures, as a result of an increased amount of link-layer transmissions, a behaviour that was found in several (but not all) transmitter-receiver pairs. The signal strength was sufficiently high for all links, therefore the impact that we observe is not connected to the decrease in signal strength at high temperatures observed in [5], [10]. The only difference among different pairs of nodes was the radio channel used for communication: each transmitter-receiver pair was assigned a different (orthogonal) channel. As the experiment was carried out in an indoor office testbed with several Wi-Fi access points, we can connect the increase of link-layer transmissions to the presence of interference in specific channels. However, we only notice an impact at high temperatures.

Further investigation led us to the identification of the problem: *the increase in link-layer transmissions was caused by a reduced efficiency of the clear channel assessment (CCA) operation at high temperatures*. Fig. 14(b) shows that the strength of the measured noise at the transmitter decreases when temperature increases (whereas it remains constant at the receiver). As highlighted in [10], the radio's received power decreases at high temperatures, and so does the measured signal strength. This implies that a source of noise in the environment will be perceived as "weaker" by a heated node, i.e., the transmitter erroneously measures a weaker noise in the environment as a result of the increased temperature. CCA algorithms are typically based on a fixed threshold T_{CCA} below which the channel is considered clear (e.g., in the CC2420 radio, T_{CCA} is set by default to -77 dBm). At high temperatures, the strength of the measured noise decreases, and there are hence higher chances that it falls below T_{CCA}, leading to a "clear channel" and a consequent packet transmission. If this happens, there is a likelihood that the transmitted packets are going to be destroyed or corrupted by interference, and our experiments confirm this observation.

We run the same experiment using Contiki's nullrdc to avoid protocol-specific behaviour and noticed that the amount of CCA failures decreases at high temperatures, leading to a substantial loss of packets. Fig. 14(b) shows that 15% of the packets were lost as a result of wrong clear channel assessments. It is important to highlight that the loss rate would be even higher in interfered scenarios, and that *protocols adapting CCA thresholds [34], [35], may be even more vulnerable to this issue*, as they would lower T_{CCA} when temperature increases as a result of the radio's decreased received power.

VII. Related Work

Traditionally, the wireless sensor networks research community relies on testbed facilities to evaluate and tune newly developed methods, protocols, and applications under realistic conditions in a cost-effective way. A large number of publicly available testbeds has been developed in the last decade, where registered users can upload the specifications of an experiment and collect traces directly via a web interface. Examples are MoteLab [12], Kansei [14], Indriya [36], TWIST [13], and NetEye [15]. The capabilities of testbeds have constantly evolved in the last years. Focus has been on reducing their management effort [37], allocating testbed resources to users that need them the most [38], accurately analysing the power consumption [39], improving data presentation and analysis [40], as well as on confederating multiple testbeds [41]. As the accuracy of a testbed experiment largely depends on how accurately environmental effects can be reproduced, recent efforts have looked at extending existing infrastructures with the emulation of environmental effects such as radio interference and mobility of nodes [42], [43], [44]. For example, in JamLab, Boano et al. [42] have added the ability to reproduce realistic interference patterns within a testbed without the need to add additional hardware equipment. In ViMobiO [43], Puccinelli and Giordano implemented a virtual mobility overlay to reproduce movement patterns of nodes during experimental evaluation.

One crucial environmental property, however, has not received significant attention in the community even though it can dramatically affect the communications between wireless sensor nodes: temperature. A few works have reported the degradation of packet loss rate [45], signal strength [6], and link quality [17] as a consequence of an increase in ambient temperature, based on observations in real-world deployments or outdoor testbed facilities. Outdoor testbeds, however, do not allow to systematically analyse the impact of temperature [46], [47]. First, meteorological conditions cannot be controlled, making it impossible to ensure repeatability across several experiments. Second, the temperature profiles that can be tested are highly specific to the deployment location and to the time of the year in which the experiment is performed.

Bannister et al. [5] have attempted to quantify the loss in received signal strength between a pair of nodes using a temperature chamber, but did not have the possibility to carry out experiments on a larger scale. Experimenting inside thermal chambers is indeed extremely costly and targets only individual components and not a network of nodes with different on-board temperatures (which is necessary to disclose limitations at the network level). TempLab aims to solve these shortcomings and provides the research community with a testbed capable of reproducing real-world temperature profiles.

In our previous work [10], we have shown that temperature affects transmitting and receiving nodes differently, and that several sensornet platforms follow a similar trend that can be captured in a simple first-order model. This work was carried out exploiting an earlier version of TempLab, but did not contain details of the testbed infrastructure, which is instead the focus of this paper.

Other studies have analysed the variations in energy consumption due to changes in temperature on sensor motes [48], and achieved an adaptive duty-cycling of sensor nodes based on underlying hardware variability [49]. Zhou and Xing [50] have designed Nemo, a power metering system for wireless sensor networks, and used it to track the sleep current consumption of motes across different temperatures by leaving motes on electric heaters and moving them outdoors to cool them down. TempLab can be a useful tool to perform similar experiments on a large-scale with higher accuracy.

VIII. Summary and Outlook

In this paper we describe TempLab, an extension for sensornet testbeds that allows to vary the on-board temperature of wireless sensor motes and study the effects of temperature variations on the network performance in a precise and repeatable fashion. We have shown that TempLab can accurately reproduce traces recorded in outdoor environments with an average error of only 0.1°C, and demonstrated that it can be a useful tool to study the significant impact of temperature on processing and protocol performance. Hence, we believe that TempLab can play an important role in studying the effects of temperature variations on the performance of wireless sensor networks, as it can reveal system limitations that would not have been visible when using existing testbed installations.

In the future we plan to automate and integrate the interpolation and association of traces into the testbed software, as these steps currently have to be performed manually using separate tools. We also plan to extend the number of nodes of our testbed (currently eighteen), and increase the capabilities of PE nodes by using an on/off wireless switch to activate or deactivate the Peltier module. Finally, although the controllers are calibrated in order not to exceed the operating range of the individual components of the sensor nodes, we will investigate if a continuous heating and cooling of the nodes accelerates the ageing process.

Acknowledgments

The research leading to these results has received funding from the European Union 7^{th} Framework Programme (FP7-ICT-2011-8) under grant agreement n° 317826 (RELYonIT). The authors would like to thank Hjalmar Wennerström for providing the temperature traces from their long-term outdoor deployment, Bernward Jelonek for his work on the control of individual Z-Wave nodes, and Felix J. Oppermann for his help in setting up the testbed.

References

[1] A. Hasler, I. Talzi, C. Tschudin, and S. Gruber. Wireless sensor networks in permafrost research – concept, requirements, implementation and challenges. In *Proc. of the 9^{th} IWSNE Worksh.*, 2008.

[2] T. Schmid. *Time in Wireless Embedded Systems*. PhD thesis, University of California, 2009.

[3] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proc. of the 6th SenSys Conf.*, 2008.

[4] C. Park, K. Lahiri, and A. Raghunathan. Battery discharge characteristics of wireless sensor nodes: An experimental analysis. In *Proc. of the 2nd SECON Conf.*, 2005.

[5] K. Bannister et al. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proc. of the 5th HotEmNets Worksh.*, 2008.

[6] C.A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. The impact of temperature on outdoor industrial sensornet applications. *IEEE Trans. Ind. Informatics*, 6(3), 2010.

[7] C.A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proc. of the 1st SensAppeal Conf.*, 2009.

[8] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of wireless sensor networks for habitat monitoring. In *Wireless Sensor Networks*, pages 399–423. 2004.

[9] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition wireless sensor networks. *Wireless Sensor Networks*, 2920:307–322, 2004.

[10] C.A. Boano et al. Hot Packets: A systematic evaluation of the effect of temperature on low power wireless transceivers. In *Proc. of the 5th ExtremeCom Conf.*, 2013.

[11] HP. *Fundamentals of Quartz Oscillators*, 1997.

[12] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a wireless sensor network testbed. In *Proc. of the 4th IPSN Conf.*, 2005.

[13] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proc. of the 2nd RealMAN Worksh.*, 2006.

[14] E. Ertin, A. Arora, R. Ramnath, M. Sridharan, and V. Kulathumani. Kansei: A testbed for sensing at scale. In *Proc. of the 5th IPSN Conf.*, 2006.

[15] X. Ju, H. Zhang, and D. Sakamuri. NetEye: A user-centered wireless sensor network testbed for high-fidelity, robust experimentation. *Int. J. Commun. Syst.*, 25(9), 2012.

[16] Abhishek Chattopadhyay. Basic RF testing of CCxxxx devices. Application Report SWRA370, 2011.

[17] H. Wennerström et al. A long-term study of correlations between meteorological conditions and 802.15.4 link performance. In *Proc. of the 10th SECON Conf.*, 2013.

[18] J. Beutel, B. Buchli, F. Ferrari, M. Keller, L. Thiele, and M. Zimmerling. X-SENSE: Sensing in extreme environments. In *Proc. of the (DATE) Conference Exhibition*, 2011.

[19] N. Thepvilojanapong, T. Ono, and Y. Tobe. A deployment of fine-grained sensor network and empirical analysis of urban temperature. *Sensors*, 10:2217–2241, 2010.

[20] Patrick L. Ryan. Radio frequency propagation differences through various transmissive materials. Master's thesis, University of North Texas, Denton, TX, USA, dec 2002.

[21] DomotiGa: Open Source Home Automation Software for Linux. *Z-Wave Technical Basics*, June 2011.

[22] Custom Thermoelectric. *ATA-050-24 Specifications Sheet*, revision 5-13-2013 edition, may 2013.

[23] F.O. Hocaoğlu, Ö.N. Gerek, and M. Kurban. Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks. *Solar Energy*, 82(8), 2008.

[24] D.E. Lane, K. Goris, and R.C.J. Somerville. Radiative transfer through broken clouds: Observations and model validation. *Journal of Climate*, 15(20), 2002.

[25] M. Rahimi, R. Baer, O.I. Iroezi, J.C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *Proc. of the 3rd SenSys Conf.*, 2005.

[26] M. Ceriotti et al. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Proc. of the 9th IPSN Conf.*, 2009.

[27] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. *IEEE Trans Biomed Eng*, 58(9), 2011.

[28] I.E. Bagci, S. Raza, T. Chung, U. Roedig, and T. Voigt. Combined secure storage and communication for the internet of things. In *Proc. of the 10th SECON Conf.*, 2013.

[29] T. Winter et al. RPL: IPv6 routing protocol for low-power and lossy networks. Technical report, IETF, 2012.

[30] N. Tsiftes, J. Eriksson, and A. Dunkels. Low-power wireless IPv6 routing with ContikiRPL. In *Proc. of the 9th IPSN Conf.*, 2010.

[31] C. Keppitiyagama et al. Temperature hints for sensornet routing. In *Proc. of the 11th SenSys Conf., poster session*, 2013.

[32] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for WSN. In *Proc. of the 1st INSS Worksh.*, 2004.

[33] P. Suriyachai, J. Brown, and U. Roedig. Time-critical data delivery in wireless sensor networks. In *Proc. of the 6th DCOSS Conf.*, 2010.

[34] M. Sha, G. Hackmann, and C. Lu. Energy-efficient low power listening for wireless sensor networks in noisy environments. In *Proc. of the 12th IPSN Conf.*, 2013.

[35] W. Yuan, J.-P. Linnartz, and I. Niemegeers. Adaptive CCA for IEEE 802.15.4 wireless sensor networks to mitigate interference. In *Proc. of the IEEE WCNC Conf.*, 2010.

[36] M. Doddavenkatappa, M. Chan, and A.L. Ananda. Indriya: A low-cost, 3D wireless sensor network testbed. In *Proc. of the 7th TridentCom Conf.*, 2011.

[37] R. Crepaldi, S. Friso, A. Harris, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi. The design, deployment, and analysis of SignetLab: A sensor network testbed and interactive management tool. In *Proc. of the 3rd TridentCom Conf.*, 2007.

[38] B.N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D.C. Parkes, J. Shneidman, A.C. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *Proc. of the 2nd EmNetS Worksh.*, 2005.

[39] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen. PowerBench: A scalable testbed infrastructure for benchmarking power consumption. In *Proc. of the 1st IWSNE Worksh.*, 2008.

[40] A.R. Dalton and J.O. Hallstrom. A file system abstraction and shell interface for a wireless sensor network testbed. In *Proc. of the 3rd TridentCom Conf.*, 2007.

[41] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer. WISEBED: An open large-scale wireless sensor network testbed. In *Proc. of the 1st Sensappeal Conf.*, 2009.

[42] C.A. Boano, T. Voigt, C. Noda, K. Römer, and M.A. Zúñiga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proc. of the 10th IPSN Conf.*, 2011.

[43] D. Puccinelli and S. Giordano. ViMobiO: Virtual mobility overlay for static sensor network testbeds. In *Proc. of the 4th EXPonWireless Worksh.*, 2009.

[44] D. Johnson, T. Stack, R. Fish, D.M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau. Mobile Emulab: A robotic wireless and sensor network testbed. In *Proc. of the 25th INFOCOM Conf.*, 2006.

[45] J. Sun and R. Cardell-Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *Proc. of the 2nd RealWSN Worksh.*, 2006.

[46] P. Dutta et al. Trio: Enabling sustainable and scalable outdoor WSN deployments. In *Proc. of the 5th IPSN Conf.*, 2006.

[47] R. Lim et al. FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proc. of the 12th IPSN Conf.*, 2013.

[48] Q. Li, M. Martins, O. Gnawali, and R. Fonseca. On the effectiveness of energy metering on every node. In *Proc. of the 10rd DCOSS Conf.*, 2013.

[49] L. Wanner, C. Apte, R. Balani, P. Gupta, and M. Srivastava. Hardware variability-aware duty cycling for embedded sensors. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(6), 2013.

[50] R. Zhou and G. Xing. Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks. In *Proc. of the 12rd IPSN Conf.*, 2013.

Paper G

C.A. Boano, K. Römer, and N. Tsiftes. **Mitigating the Adverse Effects of Temperature on Low-Power Wireless Protocols.** *In Proceedings of the 11th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS).* Philadelphia, PE, USA. October 2014.

Summary. This paper illustrates the adverse effects of temperature on communication protocols and propose techniques to increase their resilience. First, the paper experimentally shows that fluctuations of the on-board temperature of sensor nodes reduce the efficiency of data link layer protocols, leading to a substantial decrease in packet reception rate and to a considerable increase in energy consumption. Second, the paper investigates the reasons for such performance degradation, and shows that high on-board temperatures reduce the effectiveness of clear channel assessment, compromising the ability of a node to avoid collisions and to successfully wake-up from low-power mode. Third, the paper describes two mechanisms to dynamically adapt the clear channel assessment threshold to temperature changes. An extensive evaluation shows that these two mechanisms considerably increase network performance in the presence of temperature variations commonly found in real-world outdoor deployments.

My contributions. I am the main author of this paper and I conceived the idea of adapting the CCA threshold to mitigate the adverse effects of temperature variations on low-power communication protocols. I wrote the vast majority of the paper in collaboration and discussion with the co-authors, and carried out all the experiments in the evaluation section. I implemented the changes in Contiki's MAC protocols, and received help from Nicolas Tsiftes to disseminate the temperature information by piggybacking it on RPL's routing beacons. I presented the paper at MASS'14.

3. In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink.

Mitigating the Adverse Effects of Temperature on Low-Power Wireless Protocols

Carlo Alberto Boano and Kay Römer
Institute for Technical Informatics
Graz University of Technology, Austria
{cboano, roemer}@tugraz.at

Nicolas Tsiftes
SICS Swedish ICT
Stockholm, Sweden
nvt@sics.se

***Abstract*—Research and industrial installations have shown that the on-board temperature of wireless sensor nodes deployed outdoors can experience high fluctuations over time with a large variability across the network. These variations can have a strong impact on the efficiency of low-power radios and can significantly affect the operation of communication protocols, often compromising network connectivity. In this paper, we show the adverse effects of temperature on communication protocols and propose techniques to increase their resilience. First, we experimentally show that fluctuations of the on-board temperature of sensor nodes reduce the efficiency of data link layer protocols, leading to a substantial decrease in packet reception rate and to a considerable increase in energy consumption. Second, we investigate the reasons for such performance degradation, and show that high on-board temperatures reduce the effectiveness of clear channel assessment, compromising the ability of a node to avoid collisions and to successfully wake-up from low-power mode. After modelling the behaviour of radio transceivers as a function of temperature, we propose two mechanisms to dynamically adapt the clear channel assessment threshold to temperature changes, thus making data link layer protocols temperature-aware. An extensive experimental evaluation shows that our approaches considerably increase the performance of a network in the presence of temperature variations commonly found in real-world outdoor deployments, with up to 42% lower radio duty-cycle and 87% higher packet reception rate.**

Keywords*—*Clear Channel Assessment, CSMA Protocols, Outdoor Networks, Temperature Variations, Wireless Sensor Networks.

I. Introduction and Motivation

Temperature has a strong impact on the performance of wireless sensor networks. Real-world deployments have shown that the on-board temperature of wireless sensor nodes deployed outdoors can be significantly higher than air temperature [1]. Sensor nodes are indeed often exposed to direct sunlight and embedded into airtight packaging absorbing IR-radiation [2], causing the inner temperature in the casing to reach values as high as 70°C [3]. In a long-term outdoor deployment, Wennerström et al. [4] have indeed observed that the on-board temperature of a sensor node enclosed into an airtight packaging can experience variations up to 83°C across different seasons, and 56°C within 24-hours [5], with large heterogeneity across the network [6].

These temperature fluctuations can have a strong impact on clock drift, slowing down processor operations [6] and affecting time synchronization between nodes [7]; as well as on the lifetime of sensor nodes, influencing the capacity and discharge curve of batteries [8], [9] and altering the current consumption of electronic components [10], [11].
Furthermore, temperature can also drastically affect the efficiency of low-power wireless transceivers and reduce the quality of wireless links. The performance of low-power radios employed in off-the-shelf wireless sensor nodes is indeed temperature-dependent [12], with a reduction in the strength of the transmitted and received signal at high temperatures. For example, a temperature variation of 40°C can decrease the strength of the received signal by up to 6 dB, with a negative effect on the correct reception of packets [5].

To better study the impact of temperature variations on low-power wireless links and communications protocols, we have designed TempLab, a testbed infrastructure with the ability of varying the on-board temperature of sensor nodes and reproducing the temperature fluctuations found in outdoor deployments [6]. We have shown how this temperature-controlled testbed can be used to systematically analyse the performance of communication protocols, and highlighted that the latter exhibit a substantially lower efficiency at high temperatures.

In this paper, we exploit this temperature-controlled testbed to analyse in detail the performance of state-of-the-art communication protocols and to understand (i) *why* their performance decreases in the presence of temperature variations, and (ii) *how* we can mitigate the problem and improve their resilience towards temperature fluctuations. We first show experimentally that fluctuations of the on-board temperature of sensor nodes reduce the efficiency of carrier sense multiple access data link layer protocols, leading to a substantial decrease in the packet reception rate and to an increase of the energy consumption. We identify reduced effectiveness of clear channel assessment as the reason for such performance degradation, and show that this reduced effectiveness compromises the ability of a node to avoid collisions and to successfully wake-up from low-power mode. Based on these insights, we propose two mechanisms to mitigate the problem by dynamically adapting the clear channel assessment threshold to temperature changes: one based on the temperature measured locally, and one on the highest temperature measured across all neighbouring nodes. We finally show through an extensive experimental evaluation that the proposed approaches increase the robustness of existing protocols to temperature variations and significantly improve the performance also on a network level.

The contributions of this paper are hence three-fold:

- **Inefficiency of clear channel assessment.** We describe how temperature variations affect the efficiency of clear channel assessment, and show experimentally that this inefficiency compromises the operations of data link layer protocols based on carrier sense.
- **Adaptive data link layer protocols.** After modelling the behaviour of radio transceivers as a function of temperature, we implement two strategies that increase

the efficiency of clear channel assessment by making data link layer protocols temperature-aware.

- **Extensive experimental evaluation.** We show that our improved protocols sustain a significantly higher performance than existing protocols, with up to 71% lower energy consumption and 194% higher packet reception rate in the presence of temperature variations commonly found in real-world outdoor deployments.

The next section describes the impact of temperature on low-power radios, and models the attenuation of signal strength on the platform used in our experiments. Sect. III analyses the impact of temperature on data link layer protocols, and highlights the inefficiency of clear channel assessment at high temperatures. In Sect. IV we describe two mechanisms to correct this inefficiency and to make data link layer protocols temperature-aware. We evaluate the performance of our approaches in Sect. V, showing large performance improvements on a link basis and on a network level. After describing related work in Sect. VI, we conclude our paper in Sect. VII.

II. Impact of Temperature on Low-Power Radios

Experiences and reports from long-term outdoor deployments have highlighted that temperature has a strong impact on the performance of low-power radio transceivers.

Impact of temperature on link quality. Results by Bannister et al. [12] from an outdoor deployment in the Sonoran desert have revealed that an increase in temperature causes a reduction of the wireless link quality. These results were later confirmed by indoor and outdoor experiments [2], [3], and by a long-term outdoor deployment by Wennerström et al. [4] in Uppsala, Sweden. In the latter, 16 TelosB nodes equipped with the CC2420 radio were placed within each other's transmission range, and exchanged packets and recorded statistics for several months. Fig. 1 shows the data collected by two nodes in this deployment: the top figure shows the temperatures measured on-board and the air temperature recorded by a nearby weather station; the other figures show the evolution of a number of link quality metrics over time. Firstly, we can observe that the on-board temperature of the sensor nodes is significantly higher than air temperature: this is very common in outdoor deployments when nodes are enclosed into airtight packaging absorbing IR-radiation. Secondly, we can observe a clear correlation between the on-board temperature of the two nodes and the quality of their link: the higher the temperature, the lower the received signal strength indicator (RSSI) and the link quality indicator representing the chip error rate (LQI).

Dependency between temperature and signal strength. Bannister et al. [12] have shown that the attenuation in received signal strength on the CC2420 radio chip is the result of the decreased efficiency of the transmitter's power amplifier and the receiver's low-noise amplifier at high temperatures. In their experiments in a climate chamber, the authors observed a decrease of 4-5 dB in the output power of the transmitter and a drop of 3-4 dB in the received power over the temperature range 25-65 °C, for a combined effect on received signal strength of 8 dB when both transmitter and receiver are heated. We have confirmed in later experiments over a larger temperature range [5] that the relationship between temperature and signal strength attenuation is approximately linear, and that this also applies to other radio chips employed in off-the-shelf sensornet platforms. Fig. 2 shows the strength of the received signal at different temperatures between two Maxfor MTM-CM5000MSP sensor nodes (replica of TelosB motes) while the transmitter, receiver, or both transmitter and receiver nodes are heated using TempLab [6]. We can notice that the received signal strength attenuation is similar when the two nodes are heated individually (a loss of 0.08 dB/°C[1]), and about twice as high when both nodes are heated at the same time (a loss of 0.17 dB/°C). Instead, the noise floor, i.e., the received signal strength measured in absence of radio activity, exhibits a lower variability in the presence of temperature variations.

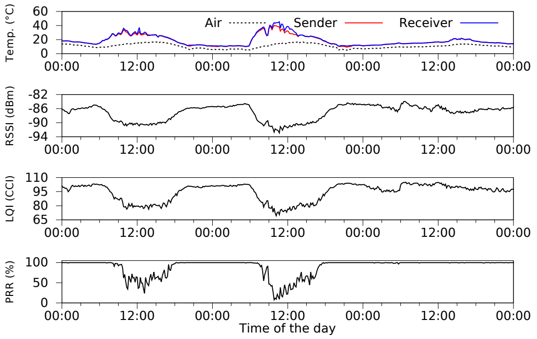


Fig. 1. High temperatures decrease the performance of low-power radios. In traces from Wennerström et al.'s outdoor deployment [4], we can observe that during daytime (when temperature is high), the received signal strength indicator (RSSI) and link quality indicator (LQI) are lower than during the night. During daytime, also the packet reception rate (PRR) is reduced.

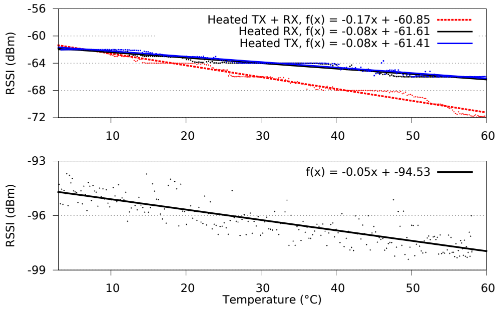


Fig. 2. Signal strength attenuation as a function of temperature. The top plot shows the received signal strength of packets while transmitter (blue), receiver (black), or both transmitter and receiver (red) are heated: the attenuation is highest when both nodes are heated at the same time. The bottom plot shows the received signal strength attenuation in absence of packet transmissions.

Impact on packet reception. The attenuation of the signal strength at high temperatures can affect the reception of packets in two ways. First, a weaker signal is more susceptible to bursts of external interference, and the probability that devices operating at higher powers (e.g., Wi-Fi access points and microwave ovens) corrupt or destroy a packet increases at high temperatures. Second, if temperature increases and the signal strength weakens to values close to the ambient RF noise (often called noise floor), the radio's ability to successfully demodulate a packet significantly decreases. When

[1] We estimate the attenuation by computing the slopes of the RSSI curve. Please note that an *exact* comparison between two curves is not possible, as RSSI readings are integer values that depend on the operation of the automatic gain controller and on the hysteresis between different gain modes [5].

this happens, a physical limit is reached: the radio cannot correctly receive (most of) the packets that were transmitted, and the connectivity of the link is irreparably compromised. This situation is captured in Fig. 1 (bottom). In Wennerström et al.'s deployment, the nodes communicate using Contiki's nullMAC, a data link layer protocol in which the radio remains active all the time and packets are transmitted without first verifying the absence of other traffic. As soon as the received signal strength weakens to values close to the noise floor in the deployment environment (≈ -94 dBm), the packet reception rate (PRR) between the two nodes drops significantly, and the link becomes almost useless during daytime.

In the next section, we focus on carrier sense multiple access data link layer protocols and show that their performance decreases significantly at high temperatures, *but not as a result of the above observations*. The vast majority of duty-cycled MAC protocols *do not* actually reach the physical limit of the radio at high temperatures, and the lower reception rates are caused by design choices that neglect the inefficiency of clear channel assessment in the presence of temperature fluctuations.

III. Impact of Temperature on CSMA Protocols

The attenuation of received signal strength at high temperatures described in Sect. II can affect two key functionalities of carrier sense multiple access (CSMA) protocols.

1) *Collision avoidance.* CSMA protocols rely on clear channel assessment (CCA) to determine whether another device is already transmitting on the same frequency channel, and defer transmissions that may otherwise collide with ongoing communications.
2) *Wake-up of nodes.* Duty-cycled protocols typically employ CCA to trigger wake-ups, i.e., to determine if a node should stay awake to receive a packet or whether it should remain in low-power mode.

CCA implementations are typically based on energy detection, i.e., on the measurement of the received signal strength and on its comparison with a given threshold. When performing energy detection using a fixed CCA threshold, it is neglected that received signal strength readings are affected by temperature, and this leads to a number of problems. First, the transmitter can erroneously measure a weaker noise in the environment as a result of the increased temperature, and generate wasteful transmissions (see Sect. III-B). Second, a receiver node may not receive a signal sufficiently strong to cause a wake-up of the radio, and constantly remain in low-power mode at high temperatures, causing the disruption of the link (see Sect. III-C). We analyse these issues in the remainder of this section, after describing how CCA is typically implemented in sensornet MAC protocols.

A. Clear Channel Assessment in Sensornet MAC Protocols

In CSMA protocols, the correct operation of clear channel assessment is fundamental to reduce the number of wasteful transmissions and to preserve the limited energy budget of the nodes in the network. The typical task of CCA is to avoid collisions, i.e., to determine whether another device is already transmitting on the same frequency channel. If there are ongoing transmissions, CSMA protocols defer transmissions using different back-off strategies [13]; otherwise the packet(s) are immediately sent. CCA is also used in low-power duty-cycled MAC protocols to trigger wake-ups, i.e., to determine if a node should remain awake to receive a packet or whether it should return in sleep mode [14]. Towards this goal, low-power MAC protocols typically perform an inexpensive CCA check and keep the transceiver on if some ongoing activity is detected on the channel [14], [15], [16].

The CCA check can be carried out using energy detection or carrier sense, as described in the IEEE 802.15.4 standard. Energy detection consists in sampling the energy level in the wireless channel and determining whether another device is already transmitting by comparing the measured signal strength with a given CCA threshold T_{CCA}. Carrier sense consists in detecting the presence of a modulated signal, irrespective of its strength. Both options can also be used at the same time: in the CC2420 transceiver, this is the default configuration.

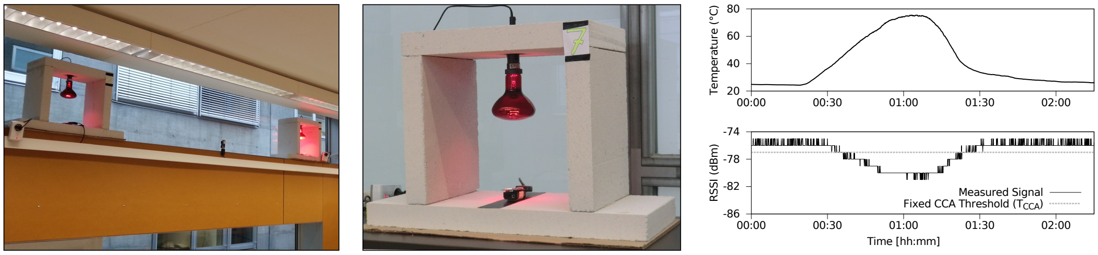
Most protocols employ fixed CCA thresholds. When using energy detection, a critical design choice is the selection of T_{CCA}. Whilst sender-initiated, duty-cycling MAC protocols such as B-MAC [14], BoX-MACs [17], and ContikiMAC [16] include energy detection as an important feature to reduce idle listening, there is not yet a widespread practice of tuning the CCA threshold at run-time in relation to the noise floor of each network deployment. Rather, the current practice is to rely on the default system settings, i.e., on a *fixed* CCA threshold, which is either set at compile-time, or left untouched so that the default value of the radio device is used instead. The IEEE 802.15.4 standard suggests to use a T_{CCA} that is at most 10 dB greater than the radio's specified receiver sensitivity. Contiki uses the default value for most hardware platforms (the CC2420's default threshold is -77 dBm), but did recently set T_{CCA} for TelosB-based platforms to -90 dBm.

B. Inefficient Collision Avoidance

When a protocol employs a fixed CCA threshold to determine whether another device is already transmitting, it essentially neglects that the received signal strength depends on the temperature. We now show experimentally that this can lead to an increase in false negatives when a transmitter is assessing the presence of a busy medium.

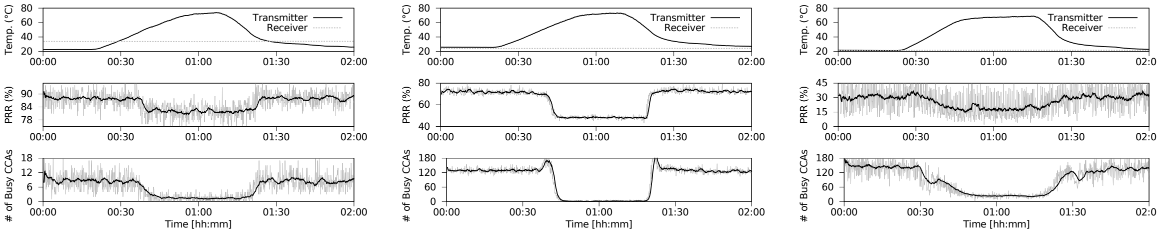
Fig. 3(a) shows an overview of our testbed, equipped with eighteen Maxfor MTM-CM5000MSP nodes. We use TempLab [6] to vary the on-board temperature of the nodes between 25 and 75°C using IR heating lamps (Fig. 3(b)). We carry out experiments consisting of several transmitter-receiver pairs running a basic Contiki application, in which the transmitter node periodically sends packets to its intended receiver and collects statistics such as the energy expenditure at the link-layer and the RF ambient noise in the radio channel. The latter is computed as the maximum of 20 consecutive RSSI readings after a packet transmission. In a first experiment in an environment rich of Wi-Fi interference, we use Contiki's nullMAC and nullRDC to avoid protocol-specific implementations and employ the CC2420's default CCA threshold (-77 dBm). Except from temperature, there is no significant change in the environment surrounding the nodes.

Fig. 3(c) shows the ambient noise captured using RSSI readings by a node in our testbed. The noise has a visible correlation with the on-board temperature of the node, and follows the attenuation described in Sect. II. We can observe



(a) Overview of our testbed infrastructure (b) IR heating lamp on top of a sensor node (c) Signal strength measured at high temperatures

Fig. 3. Overview of the testbed infrastructure used in our experiments (a) with infra-red heating lamps on top of each sensor node to control their on-board temperature (b). The received signal strength weakens at high temperatures and can cause an intersection with T_{CCA}, causing several issues (c).



(a) JamLab's emulated Wi-Fi video streaming (b) JamLab's emulated microwave oven (c) File transfer between two Wi-Fi devices

Fig. 4. Temperature affects the efficiency of collision avoidance in CSMA protocols. Our experiments in different interference scenarios show that when the received signal strength weakens to values below T_{CCA} at high temperatures, the PRR decreases, as well as the number of CCAs identifying a busy channel.

that at around 40°C, there is an intersection between the measured signal strength and the selected T_{CCA}. For temperatures lower than 40°C the measured RSSI is above T_{CCA} (and hence transmissions would be deferred); for temperatures higher than 40°C, instead, the RSSI is below T_{CCA} (and packets would be immediately sent). In other words, the MAC protocol erroneously deduces from RSSI readings obtained above 40°C that the channel is free from harmful interference. In reality, the interference in the environment is not weakened by temperature (the RSSI attenuation is only an artefact of the radio), and can still destroy transmitted packets. These erroneous clear channel assessments at high temperature may hence lead to an increase in the number of wasteful transmissions destroyed or corrupted by surrounding interference.

Fig. 4 shows the impact of erroneous clear channel assessments in the presence of different interference patterns. We use JamLab [18] to produce repeatable interference in our testbed on different channels. We emulate on one channel the interference caused by a computer streaming videos from a Wi-Fi access point, and on another channel the one caused by an active microwave oven. We also let a computer transfer large files from a nearby Wi-Fi access point using a channel that is not affected by JamLab. We then analyse how this affects the PRR on the transmitter-receiver pairs in our testbed that experienced an intersection between measured noise and T_{CCA} at different temperatures as in Fig. 3(c). We can notice that in all scenarios the PRR decreases as soon as the on-board temperature of sensor nodes increases. In the presence of Wi-Fi video streaming, the PRR of the link decreases from 88 to 81% (Fig. 4(a)), whereas in the presence of an active microwave oven the PRR decreases from 70 to 45% (Fig. 4(b)). Similarly, also the PRR in the presence of a Wi-Fi file transfer decreases from 30 to 18% at high temperatures (Fig. 4(c)). We can also notice that the decrease in PRR is correlated with a decrease in the number of clear channel assessments identifying a busy channel, i.e., with a decrease in the number of clear channel assessments that do not identify potential collisions at high temperatures. These results prove our hypothesis, and show that the intersection between the RSSI curve and the CCA threshold shown in Fig. 3(c) results in erroneous clear channel assessments leading to a decreased PRR at high temperatures.

C. Unsuccessful Wake-Up of Nodes

State-of-the-art MAC protocols often duty cycle the radio to reduce energy consumption, and employ clear channel assessment to wake-up the transceiver from sleep mode. Typically, a periodic CCA check is performed: if the channel is busy, the transceiver is kept on in order to receive the incoming packet, otherwise the radio returns to sleep mode.

High temperatures can affect the correctness of this mechanism. Imagine a sender A and a receiver B exchanging packets using a duty-cycled MAC protocol in which A sends short strobes before the actual packet (or repeatedly sends the same packet). If B receives the strobes from node A with a signal strength that is higher than T_{CCA}, it keeps its radio on and receives the payload message from A. If temperature increases, the received signal strength at node B may intersect T_{CCA} as shown in Fig. 3(c). When this happens, the transmissions from A are received with a signal strength lower than T_{CCA}, and B does not wake up to receive A's packets anymore, essentially disrupting the link. In the case shown in Fig. 3(c), the link would be disrupted for temperatures higher than 40°C, because node B would not wake up when the strength of the received signal from A decreases below T_{CCA}.

Please note that the probability that the received signal strength intersects T_{CCA} as a result of an increase in temperature can be quite high. Temperature variations can cause a

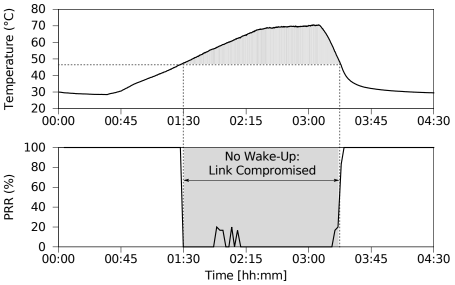


Fig. 5. Temperature can affect the wake-up mechanism in duty-cycled MAC protocols. When the strength of the received signal from a transmitter weakens at high temperatures and intersect the CCA threshold as shown in Fig. 3(c), the receiver does not wake up anymore, disrupting the link's connectivity.

signal attenuation by up to 10 dB (as shown in Fig. 2), which implies that all links in a network with an RSSI between T_{CCA} and $10+T_{CCA}$ are prone to this problem. For example, when using the CC2420 radio (T_{CCA}=−77 dBm) and transmitting at 0 dBm, the majority of nodes with a distance between 5 and 25 meters would form a link with an RSSI falling in this range [19].

We now show experimental evidence of this problem. We let several transmitter-receiver pairs of nodes communicate using ContikiMAC, Contiki's default MAC protocol in which nodes sleep most of the time and periodically wake up to check for radio activity. In ContikiMAC, the transmitter sends repeatedly the same packet until a link layer acknowledgement (ACK) is received, whereas the receiver keeps its radio on as soon as a packet transmission is detected by means of a single CCA check [16]. Packets are exchanged every 20 seconds, and ACKs are sent using CC2420's hardware support. As in the previous experiment, we use TempLab to warm-up and cool-down the on-board temperature of the nodes, emulating the daily fluctuations that can be found in real-world deployments.

Fig. 5 shows an example of link disruption caused by a receiver not waking up at high temperatures. We can notice that what was a perfect link until approximately 47°C, suddenly does not receive any packet at higher temperatures. Only once temperature decreases below 47°C, the link is restored and the node correctly receives the packets sent from the transmitter. This behaviour can significantly harm network performance, as links may disappear during the hottest times of the day, leading to high latencies, drastic topology changes, or in case no alternative paths for communication can be found, to a complete disconnection of some nodes from the network.

A receiver node could in principle detect a packet transmission using carrier sense, i.e., by identifying a valid sequence of bits without comparing if the received energy is above a given threshold. However, in off-the-shelf radio transceivers such as the CC2420, a valid sequence can be identified only prior detection and validation of the start frame delimiter. Therefore, carrier sense is ineffective when used in duty-cycled systems that periodically wake up and perform a single CCA check (in a non-duty-cycled protocol such as Contiki's nullRDC or nullMAC, instead, carrier sense would work well, as the radio remains always active). Indeed, despite the CC2420 radio uses by default a combination of carrier sense and energy detection, ContikiMAC experiences a complete loss at high temperatures that is dependent on the chosen energy detection threshold.

It is also important to highlight that selecting by default a low CCA threshold is not optimal: the lower T_{CCA} the higher the number of activities in the channel (radio interference, communications from surrounding nodes) that will trigger a wake-up and, consequently, a higher energy consumption. Indeed, selecting T_{CCA} close to the noise floor in a noisy environment, would essentially cause the radio to be almost constantly active, with a highly suboptimal energy expenditure.

IV. Designing Temperature-aware MAC Protocols

Whenever a link delivers poor performance, it is typically the network layer's task to maintain connectivity and look for alternative routes that can sustain a high delivery rate. Using link quality estimation, the network layer can indeed filter out lossy links and pick a better topology, i.e., select a network configuration that avoids links that are asymmetric or that have a signal that is too weak to communicate reliably, as well as links that are negatively affected by temperature variations. The network layer, however, can only be effective if the network is sufficiently dense to offer a high link redundancy: very often there are no available neighbours forming a link that can offer a better performance, especially in sparse networks. In such cases, the network layer is obliged to make use of lossy links, and cannot mitigate the impact of temperature variations on the lower layers of the protocol stack.

To mitigate the inefficiency of CSMA protocols at high temperatures shown in Sect. III, we hence need to tackle the problem directly at the MAC layer. A link can indeed still offer good performance if the CCA threshold is dynamically adapted to the on-board temperature variations of the nodes. In this section, we propose two alternatives to achieve this goal.

A. Predicting the Attenuation of Signal Strength

In order to dynamically adapt T_{CCA} to temperature variations, we first need to model the relationship between signal strength attenuation and temperature. In Sect. II we have shown that the latter is approximately linear, and that there are two components that need to be considered: the attenuation on the receiver side, and the one on the transmitter side.
Imagine a sender A and a receiver B exchanging packets. If the on-board temperature of B varies by ΔT_B degrees w.r.t. to an initial temperature τ, the signal will suffer an attenuation on the receiver side by $R = \beta \Delta T_B$, with ΔT_B being the difference between B's current temperature T_{now} and τ. Similarly, if the on-board temperature of A varies, its signals will be transmitted with an attenuation on the transmitter side of $T = \alpha \Delta T_A$, and B will receive a signal that is T dBm weaker. In case the temperatures of both A and B vary, the overall attenuation of the received signal strength on B is given by $R + T$. Please notice that if temperature has decreased, $\Delta T = (T_{now} - \tau)$ is negative, and R and T are not an attenuation, but instead a strengthening of the signal.

α and β are specific to the employed radio and differ only in a negligible way among different instances of the same chip. Hence, they can be characterized following the same approach shown in Sect. II: using a pair of nodes that can be heated individually, we compute the variation of signal strength on a large temperature range and derive the slope of the RSSI curves of transmitter and receiver for a given platform [5].

In the case of the Maxfor nodes employed in our experiments we derive from Fig. 2 α = β = -0.08 dB/°C. We further model the attenuation of the noise floor as $N = \gamma\Delta_T$ (which is typically smaller than R and T) and derive γ = -0.05 dB/°C.

B. Adapting the CCA Threshold at Runtime

Exploiting the above model, we can now adapt the CCA threshold at runtime. Each node needs to compute if temperature varied significantly enough to cause an attenuation of the signal strength w.r.t. an initial threshold T'_{CCA}.

As we mentioned in Sect. III, the default CCA threshold is typically fixed. However, as nodes are typically uncalibrated and have radio irregularities, a good practice would be to select $T'_{CCA} = n'_f + K$, with n'_f being the noise floor of the node, and K a constant defined at compile time. If this is the case, T'_{CCA} and n'_f are computed during the start-up phase while the node experiences an on-board temperature τ. If T'_{CCA} is fixed, we assume $\tau = 25$°C. Please note that high values of K reduce the number of activities in the channel that can trigger a wake-up of a node (minimizing energy consumption), but also reduce the number of links in the network (fewer neighbours can wake-up a node with a signal strength higher than T'_{CCA}).

Whenever temperature varies significantly, we compute the updated threshold as $T_{CCA} = T'_{CCA} + T + R$, with T and R being computed using the difference between the current temperature and τ. We apply to the computation of T_{CCA} a lower bound $n_f + C$ (with $n_f = n'_f + N$) that avoids the selection of CCA thresholds that are too close to the noise floor (this would cause the radio to continuously wake-up).

Obtaining up-to-date temperature measurements. All that is needed to adapt the threshold is hence an up-to-date information about the current on-board temperature of the nodes and the initial temperature τ stored in a 2-byte variable. Almost every off-the-shelf sensornet platform comes with an embedded temperature sensor. TelosB-based platforms carry the SHT11, a digital temperature and humidity sensor. Other platforms do not have a dedicated sensor, but several microcontrollers such as the MSP430 offer the possibility to obtain a rough estimate of the on-board temperature from a built-in temperature sensor using a specific input of the analog-to-digital converter. By periodically sampling the on-board temperature, a node can hence compare its current temperature with τ and compute Δ_T. It is important to stress that the temperature sensor should be physically on the board, to get an estimate as close as possible to the temperature of the radio chip: external sensors measuring air temperature outside the packaging may not give a sufficiently accurate estimation.

Deriving the on-board temperature of the transmitter. By knowing its current on-board temperature, a node can immediately derive N and R. If a node would adapt its CCA threshold based on this information (i.e., using $T = 0$), the inefficient collision avoidance problem at high temperatures would be solved, as well as the wake-up problem in case the temperature of the transmitter node does not vary significantly. If this node, however, receives packets sent from a node experiencing temperature fluctuations, it would need to know the temperature of the transmitter to derive T and completely mitigate the unsuccessful wake-up problem. This is a non-trivial problem, as a receiver does not necessarily know the identity of the sender by the point in time in which it performs a CCA, and as it may actually be recipient of packets sent by different nodes. Assuming that transmitter and receiver experience the same temperature variations may lead to inaccurate results: real-world deployments have shown that there can be high gradients (more than 30°C) even across spatially close nodes [6], [7] because of cloud obstructions or shade from trees or buildings in the surroundings. Similarly, setting a fixed worst-case temperature at compile-time would lead to suboptimal performance, as T_{CCA} would remain unnecessarily low most of the time.

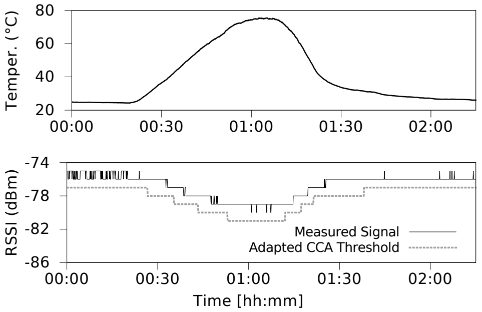
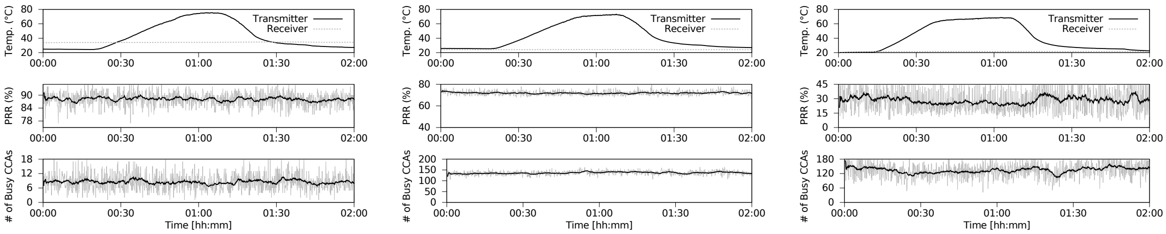


Fig. 6. Dynamic adaptation of the CCA threshold based on the temperature measured locally on the node: T_{CCA} follows the attenuation of the signal, avoiding an intersection with the RSSI curve (in contrast with Fig. 3(c)).

The information about the transmitter's temperature can actually be conveyed by the network layer, which stores a table of neighbour addresses and attributes, and can be augmented with an attribute for the latest on-board temperature of each neighbour. Modifying the network layer in this manner may not be suitable in all systems, however. Hence, we propose two different adaptation mechanisms: one that adapts T_{CCA} based only on local temperature measurements, and one that exploits a cross-layer approach to derive T.

Local adaptation. A first approach adapts T_{CCA} based on local temperature measurements only (i.e., it fixes T=0). In this case, $T_{CCA} = T'_{CCA} + R$, with a lower bound $n_f + C$. We found in our experiments that values of C below 2 dBm trigger an almost continuous wake-up of the radio, and we therefore use C=2 dBm. Fig. 6 shows the adaptation of the CCA threshold based on the algorithm detailed previously. We replicate the setup of Sect. III-B and heat a receiver node while measuring the strength of the signal in an environment rich of Wi-Fi interference. If we compare the results with the ones shown in Fig. 3(c), we can notice that the CCA threshold follows the same attenuation as the received signal, avoiding an intersection between the RSSI curve and T_{CCA}. This shows that the proposed model is sufficiently accurate to dynamically adapt T_{CCA} to local temperature changes. However, if the on-board temperature of the transmitter significantly varies, a receiver node may still experience unsuccessful wake-ups.

Cross-layer adaptation. To prevent this, we propose an approach that allows the CCA adaptation mechanism to make more informed decisions by using temperature information from the neighbours. Our cross-layer adaptation uses existing routing beacons to piggyback temperature information efficiently, and computes the maximum temperature change across all neighbours. We implement this by using RPL, the standard IPv6 routing protocol for low-power and lossy networks [20].



(a) JamLab's Wi-Fi Video Streaming (b) JamLab's Wi-Fi Active Microwave Oven (c) Real Wi-Fi File Transfer

Fig. 7. When adapting the CCA threshold based on local temperature measurements, temperature does not affect the efficiency of collision avoidance in CSMA protocols. In contrast with the results shown in Fig. 4, the PRR remains fairly constant for all interference scenarios despite temperature variations.

Whilst we have chosen RPL because it is a standard protocol and several open-source implementations exist, we also note that it would be simple to disseminate the information at the application layer, albeit with a slightly higher energy cost. We disseminate the temperature information by piggybacking it on RPL's routing beacons. RPL sends these beacons to the neighbour nodes with quickly increasing time intervals, as regulated by the Trickle algorithm [21]. Within the DODAG Information Object (DIO), there is room to embed a routing metric container object, which holds different parameters and constraints that are used to take routing decisions. Beside the metric container specified in the standard, it is possible to use implementation-defined metric containers. Hence, we make each node report its current and maximum temperature through such a metric container. Once a node receives this information in an incoming routing beacon, it stores it as an attribute in Contiki's neighbour table, from whence it can be retrieved by the CCA adaptation module to calculate the maximum temperature change in the neighbourhood.

V. Evaluation

We now evaluate the performance of our approaches experimentally. We first show that they alleviate the collision avoidance and wake-up problem in CSMA protocols. We then run a network of nodes, and show that when employing a MAC protocol with an adaptive threshold, the performance of the network significantly increases, with up to 42% lower radio duty cycle and 87% higher PRR in the presence of temperature variations commonly found in outdoor deployments.

A. Improved Collision Avoidance

In Sect. III-B we have shown that with varying on-board temperatures, a transmitter can erroneously measure a weaker noise and generate wasteful transmissions. Using the same experimental setup, we now analyse the performance of the transmitter-receiver pairs in our testbed when dynamically adapting T_{CCA} using local temperature information. We use the CC2420's default CCA threshold, i.e., $T^{'}_{CCA} = -77$ dBm and use Contiki's nullMAC and nullRDC. Fig. 7 shows the PRR experienced by the links in the same interference scenarios described in Sect. III-B (the experiments were executed back-to-back). If we compare the results with Fig. 4, we can notice that the PRR does not depend on the on-board temperature of the nodes, but remains instead fairly constant throughout the experiment. This hints that the adapted protocol is able to avoid the intersection between the RSSI curve and T_{CCA}, mitigating the collision avoidance problem.

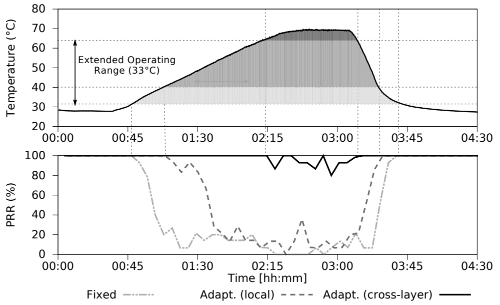
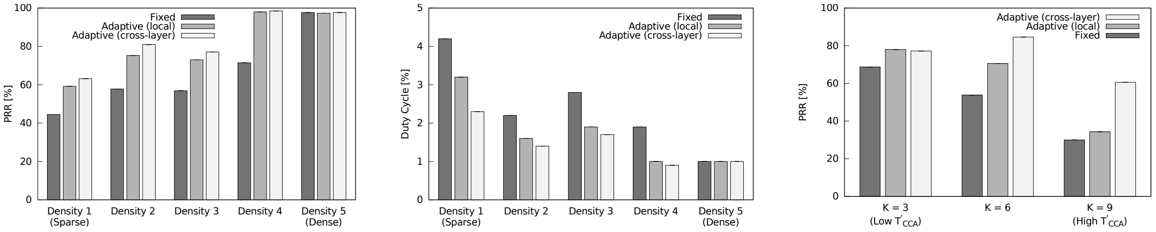


Fig. 8. Adaptive CCA thresholds alleviate significantly the wake-up problem at high temperatures. By adapting T_{CCA}, we can extend the usability of a link at much higher temperatures.

B. Improved Wake-Up Efficiency

In Sect. III-C we have shown that a receiver node exposed to temperature variations may not receive a signal sufficiently strong to cause a wake-up of the radio, and constantly remains in low-power mode, causing the disruption of the link. We employ ContikiMAC with a $T^{'}_{CCA} = n^{'}_{f} + K$ with $K = 6$ dBm and use TempLab to warm-up and cool-down the on-board temperature of both transmitter and receiver, emulating the daily fluctuations that can be found in real-world deployments. We repeat the experiments several times and run (i) an unmodified ContikiMAC using a fixed CCA threshold, (ii) an adaptive threshold based on local temperature information, and (iii) an adaptive threshold based on the information inferred from the routing layer. Fig. 8 shows the PRR on a representative link in our testbed (a similar trend was observed across all links): the adaptation of the CCA threshold can significantly alleviate the wake-up problem. When using a fixed threshold, the link starts to experience packet loss at 31°C. Instead, the link sustains 100% delivery rate up to 40°C when using local temperature information and up to 64°C when using the information inferred from the routing layer. This essentially implies that the use of a dynamic T_{CCA} extends the usability of a link to a higher temperature. It is important to highlight that the adaptation of T_{CCA} does not mitigate completely the impact of temperature. The reason lies in the selection of $T^{'}_{CCA}$: by selecting $K = 6$ dBm, the high temperature variation attenuates the signal strength by several dB, reaching the physical limit of the radio (i.e., at temperatures higher than 64°C we receive a signal strength that is too weak to be successfully demodulated). Hence, the higher is K, the higher can be the performance gain compared to a protocol using a fixed CCA threshold.



(a) PRR as a function of network density

(b) Duty cycle as a function of network density

(c) PRR as a function of $T^{'}_{CCA}$

Fig. 9. Network performance when using fixed and adaptive CCA thresholds as a function of network density (a, b), and $T^{'}_{CCA}$ (c).

C. Performance on a Network Level

We now present results obtained running a data-collection protocol on several networks, and show the benefits of using dynamically adapted CCA thresholds in the presence of temperature variations. We use RPL in our testbed deployed in a 55 m^2 room: we select one node as a sink, and we create five different network densities by using only a portion of the nodes: 5, 7, 9, 11, and 13 nodes, respectively. By varying the density from roughly one node every 11 m^2 to a node every 4 m^2, we can see largely different impacts on a network level, as the ability of the network layer to select alternative links is constrained. Using the same temperature profiles and setup as in the previous example (all nodes use a transmission power of -25 dBm), we carry out experiments with ContikiMAC using: (i) a fixed CCA threshold, (ii) an adaptive threshold based on local temperature information, and (iii) an adaptive threshold using the information inferred from the network layer.

Our results indicate that temperature strongly affects network performance, especially in sparse networks. Fig. 9(a) shows that if the network is dense, the routing layer can mitigate the impact of temperature and sustain a high PRR even with a MAC protocol employing a fixed CCA threshold. The less dense the network is, the higher becomes the impact of temperature on a protocol using a fixed threshold, with the average PRR in the network dropping below 50%. Instead, when using adaptive thresholds, the network sustains higher reception rates in sparse networks (from 44 to 63%, and from 57 to 81% in the two sparsest configurations), with the highest PRR recorded when using the information inferred from the routing layer in line with the experiments in Sect. V-B.

We further analyse the energy-efficiency of the different approaches by comparing the average radio duty cycle in the network. Fig. 9(b) shows that adaptive CCA thresholds sustain significantly lower duty cycles, as a result of a reduced number of retransmission attempts and wasteful transmissions. In the sparsest network configuration, the duty cycle drops from 4.2% to 3.2% in the case of local temperature information and to 2.3% when using the temperature inferred from the routing layer. The latter corresponds to a 55% higher energy-efficiency than when using a fixed threshold. With denser networks the duty cycle decreases, as the network layer can select alternative links and seamlessly mitigate the impact of temperature.

Fig. 9(c) shows the role of the initial CCA threshold $T^{'}_{CCA}$ in a network with a density of one node every 8 m^2. We set $T^{'}_{CCA} = n^{'}_{f} + K$ using different K values, and show that the higher K is, the higher are the performance improvements introduced by the adaptive approaches. This is the result of the observation made in Sect. V-B: the higher K is, the more the usability of a link can be extended at high temperatures.

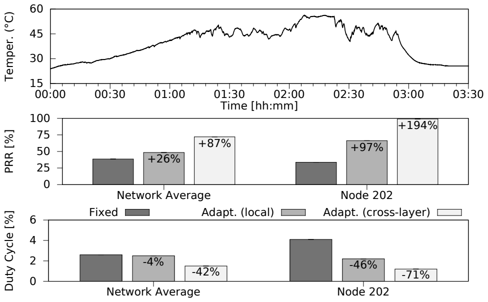


Fig. 10. Regeneration of a real-world outdoor temperature trace and impact on PRR and duty cycle on a network level and on a single node.

We finally use TempLab to time-lapse a 24-hours trace recorded in an outdoor deployment [4], and see what is the impact in a network with a density of one node every 8 m^2 when using $T^{'}_{CCA} = n^{'}_{f} + 6$ dBm. The results show that the adaptive approaches that we proposed significantly improve performance, both on a link basis and on a network level. Fig. 10 shows that the network sustains up to 42% lower radio duty cycle and 87% higher PRR in the presence of temperature variations commonly found in outdoor deployments, and that a single link may experience even up to 71% lower duty cycle and 194% higher packet reception rate.

VI. Related Work

Several outdoor deployments and experimental studies have highlighted the impact of temperature on the quality of communications in wireless sensor networks. Bannister et al. [12] have reported that high temperatures can decrease the strength of the wireless signal. Wennerström et al. [4] have found experimental evidence of this problem on a long-term outdoor deployment. Boano et al. [3] have shown that the transmission power of communications at low temperatures can be safely decreased without deteriorating the performance of the network, and have precisely characterized the attenuation in received signal strength on different platforms [5]. All these works, however, simply report the degradation of the wireless signal as a consequence of an increase in temperature and do not provide a deeper analysis of what the implications are on communication protocols when operating a network outdoors.

Keppitiyagama et al. [22] have presented a poster showing that network protocols are affected by temperature and proposed to enhance them with temperature hints. In our earlier work, we have presented TempLab, a testbed infrastructure to study the impact of temperature on communication protocols [6], and used it to confirm the low performance at high temperatures. In this work, we exploit this testbed infrastructure to analyse *why* communication protocols are affected and propose *how* to mitigate the problems by dynamically adapting the CCA threshold to temperature variations.

Several works have suggested the use of adaptive CCA thresholds in the context of interference mitigation. Bertocco et al. [23] have provided hints for an optimal threshold selection in the presence of in-channel additive white Gaussian noise interference. Yuan et al. [24] have proposed to adjust the CCA threshold in the presence of heavy interference to reduce the amount of discarded packets due to channel access failures. Xu et al. [25] have designed a mechanism that dynamically adjusts the CCA threshold to enable concurrent transmissions on adjacent non-orthogonal channels and achieve high throughput.

Sha et al. [26] have studied the effects of the CCA threshold setting in noisy environments, and shown that interference can increase the number of false wake-ups in low-power-listening MAC protocols. To remedy this problem, they have proposed AEDP, an adaptive protocol that adjusts the CCA threshold in response to changes of ETX. While we share the idea that the CCA threshold cannot be set to an arbitrary value at compile-time, there are considerable differences with our work. First, AEDP is designed to achieve a desired performance in noisy environments and does not take into account the role of temperature. This may lead to problems, as AEDP requires an estimate of the noise floor and of the average RSSI value of all incoming links, which may change as temperature changes. Second, AEDP does not require a temperature model to adapt the CCA threshold, but instead requires information of observed interference in recent packet transmission attempts. In event-based networks, the reliance on ETX values may be a problem since packet transmissions are sparse.

An alternative approach to mitigate the impact of temperature may consist in increasing the transmission power at high temperatures, as suggested by the data-sheets of some radio chips. Although this would lead to an increased energy-consumption, it may simply not be possible: a node could already be using its highest power level. Furthermore, increasing the power based on the local temperature would only make the transmitted signal stronger, but would not solve the attenuation on the receiver side. Hence, our approach based on the signal strength attenuation modelling is more generic and effective.

VII. Conclusions

The central tenet of our study is that temperature variations affect the efficiency of clear channel assessment and may compromise the operations of data link layer protocols based on carrier sense. We have shown that a reduced effectiveness of CCA at high temperatures compromises the ability of a node to avoid collisions and to successfully wake up from low-power mode. We have designed and evaluated two mechanisms to mitigate the problem by dynamically adapting the CCA threshold to temperature changes: one based on the temperature measured locally, and one on the highest temperature measured across all neighbouring nodes. Through an extensive experimental evaluation, we have shown that the proposed approaches increase the robustness of existing protocols to temperature variations and significantly improve the performance both on a link basis and on a network level.

References

[1] R. Szewczyk et al., “Lessons from a sensor network expedition,” *Wireless Sensor Networks*, vol. 2920, 2004.

[2] C. A. Boano et al., “Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance,” in *Proc. of the* 1^{st} *SensAppeal*, 2009.

[3] C. A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt, “The impact of temperature on outdoor industrial sensornet applications,” *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, 2010.

[4] H. Wennerström, F. Hermans, O. Rensfelt, C. Rohner, and L.-A. Nordén, “A long-term study of correlations between meteorological conditions and 802.15.4 link performance,” in *Proc. of the* 10^{th} *SECON*, 2013.

[5] C. A. Boano et al., “Hot Packets: A systematic evaluation of the effect of temperature on low power wireless transceivers,” in *Proc. of the* 5^{th} *ExtremeCom*, 2013.

[6] C. A. Boano, M. Zúñiga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Römer, “TempLab: A testbed infrastructure to study the impact of temperature on WSNs,” in *Proc. of the* 13^{th} *IPSN*, 2014.

[7] J. Beutel, B. Buchli, F. Ferrari, M. Keller, L. Thiele, and M. Zimmerling, “X-SENSE: Sensing in extreme environments,” in *Proc. of DATE*, 2011.

[8] C. Park, K. Lahiri, and A. Raghunathan, “Battery discharge characteristics of wireless sensor nodes: An experimental analysis,” in *Proc. of the* 2^{nd} *SECON*, 2005.

[9] W. Guo, W. M. Healy, and M. Zhou, “Experimental study of the thermal impacts on wireless sensor batteries,” in *Proc. of the* 10^{th} *ICNSC*, 2013.

[10] Q. Li, M. Martins, O. Gnawali, and R. Fonseca, “On the effectiveness of energy metering on every node,” in *Proc. of the* 10^{th} *DCOSS*, 2013.

[11] L. Wanner, C. Apte, R. Balani, P. Gupta, and M. Srivastava, “Hardware variability-aware duty cycling for embedded sensors,” *IEEE Transactions on VLSI Systems*, vol. 21, no. 6, 2013.

[12] K. Bannister, G. Giorgetti, and S. K. Gupta, “Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization,” in *Proc. of the* 5^{th} *HotEmNets*, 2008.

[13] C. A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M. Zúñiga, “Making sensornet MAC protocols robust against interference,” in *Proc. of the* 7^{th} *EWSN*, 2010.

[14] J. Polastre, J. Hill, and D. E. Culler, “Versatile low-power media access for wireless sensor networks,” in *Proc. of the* 2^{nd} *SenSys*, 2004.

[15] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proc. of the* 4^{th} *SenSys*, 2006.

[16] A. Dunkels, “The contikimac radio duty cycling protocol,” Swedish Institute of Computer Science, Tech. Rep. T2011:13, 2011.

[17] D. Moss and P. Levis, “BoX-MACs: Exploiting physical and link layer boundaries in low-power networking,” Stanford University, Tech. Rep. SING-08-00, 2008.

[18] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, “JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation,” in *Proc. of the* 10^{th} *IPSN*, 2011.

[19] J. Polastre, R. Szewczyk, and D. Culler, “Telos: Enabling ultra-low power wireless research,” in *Proc. of the* 4^{th} *IPSN*, 2005.

[20] T. Winter et al., “RPL: IPv6 routing protocol for low-power and lossy networks,” IETF, Tech. Rep., 2012.

[21] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The trickle algorithm,” IETF, Tech. Rep., 2011.

[22] C. Keppitiyagama, N. Tsiftes, C. A. Boano, and T. Voigt, “Temperature hints for sensornet routing,” in *Proc. of the* 11^{th} *SenSys*, 2013.

[23] M. Bertocco, G. Gamba, and A. Sona, “Experimental optimization of CCA thresholds in wireless sensor networks in the presence of interference,” in *Proc. of the IEEE EMC*, 2007.

[24] W. Yuan, J.-P. M. Linnartz, and I. G. M. M. Niemegeers, “Adaptive CCA for IEEE 802.15.4 wireless sensor networks to mitigate interference,” in *Proc. of IEEE WCNC*, 2010.

[25] X. Xu, J. Luo, and Q. Zhang, “Design of non-orthogonal multi-channel sensor networks,” in *Proc. of the* 30^{th} *ICDCS*, 2010.

[26] M. Sha, G. Hackmann, and C. Lu, “Energy-efficient low power listening for wireless sensor networks in noisy environments,” in *Proc. of the* 12^{th} *IPSN*, 2013.

Bibliography

[1] CONET integrated testbed. https://conet.us.es/cms/. Last visited: 30.06.2014.

[2] The Contiki Projects Community. http://sourceforge.net/projects/contikiprojects/. Last visited: 30.06.2014.

[3] GINSENG – performance control in wireless sensor networks. http://www.ict-ginseng.eu, 2008–2011. Last visited: 30.06.2012.

[4] WSAN4CIP – securing tomorrow's critical infrastructures. http://www.wsan4cip.eu/, 2008–2011. Last visited: 30.06.2014.

[5] The CREW project: Cognitive radio experimentation world. http://www.crew-project.eu/, 2010–2015. Last visited: 30.06.2014.

[6] RELYonIT – research by experimentation for dependability on the Internet of Things. http://www.relyonit.eu, 2012–2015. Last visited: 30.06.2014.

[7] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, L. Luo, S. Son, J. Stankovic, R. Stoleru, and A. Wood. EnviroTrack: Towards an environmental computing paradigm for distributed sensor networks. In *Proceedings of the* 24*th* *International Conference on Distributed Computing Systems (ICDCS)*, pages 582–589, Mar. 2004.

[8] Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *Proceedings of the* 10*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 246–257, Apr. 2011.

[9] N. Ahmed, S. S. Kanhere, and S. Jha. Mitigating the effect of interference in wireless sensor networks. In *Proceedings of the* 35*th* *IEEE International Conference on Local Computer Networks (LCN)*, pages 160–167, Oct. 2010.

[10] M. H. Alizai, O. Landsiedel, J. Ágila Bitsch Link, S. Götz, and K. Wehrle. Bursty traffic over bursty links. In *Proceedings of the* 7*th* *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 71–84, Nov. 2009.

[11] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori. Performance measurements of motes sensor networks. In *Proceedings of the* 7*th* *ACM International*

Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pages 174–181, Oct. 2004.

[12] G. Anastasi, O. Farruggia, G. Lo Re, and M. Ortolani. Monitoring high-quality wine production using wireless sensor networks. In *Proceedings of the* 42^{nd} *International Conference on System Sciences (HICSS)*, pages 1–7, Jan. 2009.

[13] J. Ansari, T. Ang, and P. Mähönen. Spectrum agile medium access control protocol for wireless sensor networks. In *Proceedings of the* 7^{th} *IEEE International Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, June 2010.

[14] J. Ansari, T. Ang, and P. Mähönen. WiSpot: Fast and reliable detection of wi-fi networks using IEEE 802.15.4 radios. In *Proceedings of the* 9^{th} *ACM International Workshop on Mobility Management and Wireless Access (MobiWac)*, pages 35–44, Oct. 2011.

[15] T. Antoine-Santoni, J.-F. Santucci, E. De Gentili, X. Silvani, and F. Morandini. Performance of a protected wireless sensor network in a fire: Analysis of fire spread and data transmission. *Sensors*, 9(8):5878–5893, July 2009.

[16] S. Arkoulis, D.-E. Spanos, S. Barbounakis, A. Zafeiropoulos, and N. Mitrou. Cognitive radio-aided wireless sensor networks for emergency response. *Measurement Science and Technology*, 21(12), Dec. 2010.

[17] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, Jan. 2004.

[18] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch. Information centric networking in the IoT: Experiments with NDN in the wild. In *Proceedings of the* 1^{st} *ACM Conference on Information-Centric Networking (ICN)*, pages 77–86, Sept. 2014.

[19] N. Baccour, M. B. Jamâa, D. do Rosário, A. Koubâa, H. Youssef, M. Alves, and L. B. Becker. A testbed for the evaluation of link quality estimators in wireless sensor networks. In *Proceedings of the International Workshop on Future Trends on Ad-hoc and Sensor Networks (IEEE FT-ASN)*, pages 1–8, May 2010.

[20] N. Baccour, A. Koubâa, L. Mottola, H. Youssef, M. A. Zúñiga, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34:1–34:33, Nov. 2012.

[21] J. Balendonck, J. Hemming, B. A. J. van Tuijl, L. Incrocci, A. Pardossi, and P. Marzialetti. Sensors and wireless sensor networks for irrigation management under deficit conditions (FLOW-AID). In *Proceedings of the International Conference on Agricultural Engineering and Agricultural & Biosystems Engineering for a Sustainable World (AgEng)*, June 2008.

[22] K. Bannister. Impacts of thermal reduction in transceiver performance on outdoor sensing networks. Master's thesis, Arizona State University, Phoenix, AZ, USA, May 2009.

[23] K. Bannister, G. Giorgetti, and S. K. Gupta. Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization. In *Proceedings of the* 5^{th} *International Workshop on Embedded Networked Sensors (HotEmNets)*, June 2008.

[24] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the* 6^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 43–56, Nov. 2008.

[25] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. SensorScope: Out-of-the-box environmental monitoring. In *Proceedings of the* 7^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 332–343, Apr. 2008.

[26] R. Beckwith, D. Teibel, and P. Bowen. Unwired wine: Sensor networks in vineyards. In *Proceedings of IEEE Sensors*, pages 561–564, Oct. 2004.

[27] M. Bertocco, G. Gamba, and A. Sona. Experimental optimization of CCA thresholds in wireless sensor networks in the presence of interference. In *Proceedings of the IEEE Europe the Workshop on ElectroMagnetic Compatibility (EMC)*, June 2007.

[28] M. Bertocco, G. Gamba, and A. Sona. Is CSMA/CA really efficient against interference in a wireless control system? an experimental answer. In *Proceedings of the* 13^{th} *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 885–892, Sept. 2008.

[29] J. Beutel, B. Buchli, F. Ferrari, M. Keller, L. Thiele, and M. Zimmerling. X-SENSE: Sensing in extreme environments. In *Proceedings of the Conference on Design, Automation, and Test in Europe (DATE)*, pages 1–6, Mar. 2011.

[30] J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, I. Talzi, L. Thiele, C. Tschudin, M. Woehrle, and M. Yuecel. Permadaq: A scientific instrument for precision sensing and data recovery in environmental extremes. In *Proceedings of the* 8^{th} *International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 265–276, Apr. 2009.

[31] J. Beutel, K. Römer, M. Ringwald, and M. Woehrle. Deployment techniques for sensor networks. In *Sensor Networks*, Signals and Communication Technology, pages 219–248. Springer Berlin Heidelberg, 2009.

[32] C. A. Boano. Application support design for wireless sensor networks. Master's thesis, Politecnico di Torino and Kungliga Tekniska Högskolan, Turin, Italy, and Stockholm, Sweden, Mar. 2009.

[33] C. A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt. Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and ATEX-compliance. In *Proceedings of the* 1^{st} *International Conference on Sensor Networks Applications, Experimentation and Logistics (SENSAPPEAL)*, pages 159–176. Springer Berlin Heidelberg, Sept. 2009.

[34] C. A. Boano, J. Brown, N. Tsiftes, U. Roedig, and T. Voigt. The impact of temperature on outdoor industrial sensornet applications. *IEEE Transactions on Industrial Informatics*, 6(3):451–459, Aug. 2010.

[35] C. A. Boano, Z. He, Y. Li, T. Voigt, M. A. Zúñiga, and A. Willig. Controllable radio interference for experimental and testing purposes in wireless sensor networks. In *Proceedings of the* 4^{th} *International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, pages 865–872. IEEE, Oct. 2009.

[36] C. A. Boano, M. Lasagni, and K. Römer. Non-invasive measurement of core body temperature in marathon runners. In *Proceedings of the* 10^{th} *IEEE International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 274–279. IEEE, May 2013.

[37] C. A. Boano, F. J. Oppermann, and K. Römer. The use of body sensor networks in clinical settings and medical research. In *Sensor Networks for Sustainable Development*, chapter 11, pages 215–252. CRC Press, July 2014.

[38] C. A. Boano and K. Römer. External radio interference. In *Radio Link Quality Estimation in Low-Power Wireless Networks*, SpringerBriefs in Electrical and Computer Engineering - Cooperating Objects, pages 21–63. Springer International Publishing, July 2013.

[39] C. A. Boano, K. Römer, Z. He, T. Voigt, M. A. Zúñiga, and A. Willig. Generation of controllable radio interference for protocol testing in wireless sensor networks. In *Proceedings of the* 7^{th} *ACM International Conference on Embedded Networked Sensor Systems (SenSys), demo session*, pages 301–302. ACM, Nov. 2009.

[40] C. A. Boano, K. Römer, F. Österlind, and T. Voigt. Realistic simulation of radio interference in COOJA. In *Adjunct Proceedings of the* 8^{th} *European Conference on Wireless Sensor Networks (EWSN), demo session*, pages 36–37, Feb. 2011.

[41] C. A. Boano, K. Römer, and N. Tsiftes. Mitigating the adverse effects of temperature on low-power wireless protocols. In *Proceedings of the* 11^{th} *International Conference on Mobile Ad hoc and Sensor Systems (MASS)*. IEEE, Oct. 2014.

[42] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proceedings of the* 10^{th} *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 175–186. IEEE, Apr. 2011.

[43] C. A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Römer, and M. A. Zúñiga. Making sensornet MAC protocols robust against interference. In *Proceedings of the* 7^{th} *European Conference on Wireless Sensor Networks (EWSN)*, pages 272–288. Springer Berlin Heidelberg, Feb. 2010.

[44] C. A. Boano, H. Wennerström, M. A. Zúñiga, J. Brown, C. Keppitiyagama, F. J. Oppermann, U. Roedig, L.-Å. Nordén, T. Voigt, and K. Römer. Hot Packets: A systematic evaluation of the effect of temperature on low power wireless transceivers. In *Proceedings of the* 5^{th} *Extreme Conference on Communication (ExtremeCom)*, pages 7–12. ACM, Aug. 2013.

[45] C. A. Boano, M. A. Zúñiga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Römer. TempLab: A testbed infrastructure to study the impact of temperature on wireless sensor networks. In *Proceedings of the* 13^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 95–106. IEEE, Apr. 2014.

[46] C. A. Boano, M. A. Zúñiga, K. Römer, and T. Voigt. JAG: Reliable and predictable wireless agreement under external radio interference. In *Proceedings of the* 33^{rd} *IEEE International Real-Time Systems Symposium (RTSS)*, pages 315–326. IEEE, Dec. 2012.

[47] J. Brown, U. Roedig, C. A. Boano, and K. Römer. Estimating packet reception rate in noisy environments. In *Proceedings of the* 9^{th} *International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*. IEEE, Sept. 2014.

[48] D. Brunelli, D. Balsamo, G. Paci, and L. Benini. Temperature compensated time synchronisation in wireless sensor networks. *IEEE Electronics Letters*, 48(16):1026–1028, Aug. 2012.

[49] 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* 4^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 307–320, Nov. 2006.

[50] M. Buevich, D. Schnitzer, T. Escalada, A. Jacquiau-Chamski, and A. Rowe. A system for fine-grained remote monitoring, control and pre-paid electrical service in rural microgrids. In *Proceedings of the* 13^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12, Apr. 2014.

[51] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, Mar. 2004.

[52] B. Capsuto and J. Frolik. A system to monitor signal fade due to weather phenomena for outdoor sensor systems. In *Proceedings of the* 5^{th} *International Conference on Information Processing in Sensor Networks (IPSN), demo session*, Apr. 2006.

[53] J. M. Castillo-Secilla, J. M. Palomares, and J. Olivares. Temperature-compensated clock skew adjustment. *Sensors*, 13(8):10981–11006, Aug. 2013.

[54] M. Ceriotti, M. Chini, A. L. Murphy, G. P. Picco, F. Cagnacci, and B. Tolhurst. Motes in the jungle: Lessons learned from a short-term wsn deployment in the ecuador cloud forest. In *Proceedings of the* 4^{th} *International Conference on Real-World Wireless Sensor Networks (REALWSN)*, pages 25–36, Dec. 2010.

[55] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, Ştefan Gună, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. Is there light at the ends of the tunnel? wireless sensor networks for adaptive lighting in road tunnels. In *Proceedings of the* 10^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 187–198, Apr. 2011.

[56] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, Ştefan Gună, M. Corrà, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Proceedings of the* 8^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 277–288, Apr. 2009.

[57] T. Chang and Q. Wang. Adaptive compensation for time-slotted synchronization in wireless sensor network. *International Journal of Distributed Sensor Networks (IJDSN)*, 2014(540397):1–9, Apr. 2014.

[58] A. Chattopadhyay. Basic RF testing of CCxxxx devices. Application Report SWRA370, Aug. 2011.

[59] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer. WISEBED: An open large-scale wireless sensor network testbed. In *Proceedings of the* 1^{st} *International Conference on Sensor Networks Applications, Experimentation and Logistics (Sensappeal)*, pages 68–87, Sept. 2009.

[60] O. Chipara, C. Lu, T. C. Bailey, and R. Gruia-Catalin. Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit. In *Proceedings of the* 8^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 155–168, Nov. 2010.

[61] K. R. Chowdhury and I. F. Akyildiz. Interferer classification, channel selection and transmission adaptation for wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1–5, June 2009.

[62] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *Proceedings of the* 2^{nd} *IEEE Workshop on Embedded Networked Sensors (EmNetS)*, pages 19–28, May 2005.

[63] R. Crepaldi, S. Friso, A. Harris, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi. The design, deployment, and analysis of SignetLab: A sensor network testbed and interactive management tool. In *Proceedings of the* 3^{rd} *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, pages 1–10, May 2007.

[64] Custom Thermoelectric. *ATA-050-24 Specifications Sheet*, revision 5-13-2013 edition, May 2013.

[65] A. R. Dalton and J. O. Hallstrom. A file system abstraction and shell interface for a wireless sensor network testbed. In *Proceedings of the* 3^{rd} *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, pages 1–10, May 2007.

[66] S. Dawson-Haggerty, S. Lanzisera, J. Taneja, R. Brown, and D. Culler. @scale: Insights from a large, long-lived appliance energy WSN. In *Proceedings of the* 11*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 37–48, Apr. 2012.

[67] M. Doddavenkatappa, M. Chan, and A. Ananda. Indriya: A low-cost, 3D wireless sensor network testbed. In *Proceedings of the* 7*th* *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, pages 302–316, Apr. 2011.

[68] DomotiGa: Open Source Home Automation Software for Linux. *Z-Wave Technical Basics*, June 2011. Last visited: 30.06.2014.

[69] D. M. Doolin and N. Sitar. Wireless sensors for wildfire monitoring. In *Proceedings of the SPIE Symposium on Smart Structures & Materials*, volume 5765, pages 477–484, May 2005.

[70] P. Du and G. Roussos. Adaptive channel hopping for wireless sensor networks. In *Proceedings of the IEEE International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, pages 19–23, Oct. 2011.

[71] A. Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report T2011:13, Swedish Institute of Computer Science, Kista, Sweden, Dec. 2011.

[72] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the* 1*st* *International Workshop on Embedded Networked Sensors (EmNetS)*, pages 455–462, Nov. 2004.

[73] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the* 4*th* *International Workshop on Embedded Networked Sensors (EmNetS)*, pages 28–32, June 2007.

[74] S. Duquennoy, F. Österlind, and A. Dunkels. Lossy links, low power, high throughput. In *Proceedings of the* 9*th* *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 12–25, Nov. 2011.

[75] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the* 8*th* *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–14, Nov. 2010.

[76] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: Enabling sustainable and scalable outdoor WSN deployments. In *Proceedings of the* 5*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 407–415, Apr. 2006.

[77] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the* 5*th* *International Symposium on Operating Systems Design and Implementation (OSDI)*, pages 147–163, Dec. 2002.

[78] E. Ertin, A. Arora, R. Ramnath, M. Sridharan, and V. Kulathumani. Kansei: A testbed for sensing at scale. In *Proceedings of the* 5*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 339–406, Apr. 2006.

[79] D. Evans. The Internet of Things – how the next evolution of the internet is changing everything. Technical report, Cisco Internet Business Solutions Group (IBSG), Apr. 2011.

[80] F. Fabbri, M. A. Zúñiga, D. Puccinelli, and P. J. Marrón. On the optimal blacklisting threshold for link selection in wireless sensor networks. In *Proceedings of the* 9*th* *European Conference on Wireless Sensor Networks (EWSN)*, pages 147–162, Feb. 2012.

[81] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the* 10*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 73–84, Apr. 2011.

[82] H. Fotouhi, M. A. Zúñiga, M. Alves, A. Koubâa, and P. J. Marrón. smart-HOP: a reliable handoff mechanism for mobile wireless sensor networks. In *Proceedings of the* 9*th* *European Conference on Wireless Sensor Networks (EWSN)*, pages 131–146, Feb. 2012.

[83] C. Frank and K. Römer. Algorithms for generic role assignment in wireless sensor networks. In *Proceedings of the* 3*rd* *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 230–242, Nov. 2005.

[84] K. Garg, A. Förster, D. Puccinelli, and S. Giordano. Towards realistic and credible wireless sensor network evaluation. In *Proceedings of the* 3*rd* *International Conference on Ad Hoc Networks (ADHOCNETS)*, pages 49–64, Sept. 2011.

[85] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. E. Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of the* 24*th* *International Conference on Programming Language Design and Implementation (PLDI)*, pages 1–11, June 2003.

[86] J. A. Gay-Fernandez, M. G. Sánchez, I. Cuinas, A. V. Alejos, J. G. Sanchez, and J. L. Miranda-Sierra. Propagation analysis and deployment of a wireless sensor network in a forest. *Progress In Electromagnetics Research*, 106:121–145, July 2010.

[87] G. Giorgetti, A. Cidronali, S. K. Gupta, and G. Manes. Exploiting low-cost directional antennas in 2.4 ghz IEEE 802.15.4 wireless sensor networks. In *Proceedings of the* 10*th* *European Conference on Wireless Technologies (ECWT)*, pages 217–220, Oct. 2007.

[88] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the* 7*th* *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–14, Nov. 2009.

[89] A. Gonga, O. Landsiedel, P. Soldati, and M. Johansson. Revisiting multi-channel communication to mitigate interference and link dynamics in wireless sensor networks. In *Proceedings of the* 8*th* *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 186–193, May 2012.

[90] J. Gray. Notes on data base operating systems. In *Operating Systems, an Advanced Course*, pages 393–481, 1978.

[91] W. Guo, W. M. Healy, and M. Zhou. Experimental study of the thermal impacts on wireless sensor batteries. In *Proceedings of the* 10*th* *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 430–435, Apr. 2013.

[92] F. Hackbarth, T. Meyerhoff, H. Sauff, B. T. Bradford, L. Torres, H. Klimek, B. Gressmann, C. Renner, M. Stemick, C. Weyer, and S. Georgi. SomSeD: The evolution of an experimental wireless sensor network towards a research platform. In *Proceedings of the* 8*th* *GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, pages 27–30, Aug. 2009.

[93] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the* 2*nd* *International Workshop on Multi-hop Ad-Hoc Networks: from Theory to Reality (REALMAN)*, pages 63–70, May 2006.

[94] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen. PowerBench: A scalable testbed infrastructure for benchmarking power consumption. In *Proceedings of the* 1*st* *International Workshop on Sensor Network Engineering (IWSNE)*, pages 37–44, June 2008.

[95] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *Proceedings of the* 4*th* *International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 28–41, June 2006.

[96] A. Hasler, I. Talzi, C. Tschudin, and S. Gruber. Wireless sensor networks in permafrost research – concept, requirements, implementation and challenges. In *Proceedings of the* 9*th* *International Conference on Permafrost (NICOP)*, June 2008.

[97] J.-H. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks. In *Proceedings of the* 6*th* *European Conference on Wireless Sensor Networks (EWSN)*, pages 17–32, Feb. 2009.

[98] J.-H. Hauer, A. Willig, and A. Wolisz. Mitigating the effects of RF interference through RSSI-based error recovery. In *Proceedings of the* 7*th* *European Conference on Wireless Sensor Networks (EWSN)*, pages 224–239, Feb. 2010.

[99] H. Hellbrück, M. Pagel, A. Kröller, D. Bimschas, D. Pfisterer, and S. Fischer. Using and operating wireless sensor network testbeds with WISEBED. In *Proceedings of the* 10*th* *Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net)*, pages 171–178, June 2011.

[100] Hewlett Packard. *Fundamentals of Quartz Oscillators*, May 1997.

[101] J. L. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, Nov. 2002.

[102] M. A. Hossian, A. Mahmood, and R. Jäntti. Channel ranking algorithms for cognitive coexistence of IEEE 802.15.4. In *Proceedings of the* 20^{th} *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 112–116, Sept. 2009.

[103] G. J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *Proceedings of the* 18^{th} *IEEE International Conference on Network Protocols (ICNP)*, pages 305–314, Oct. 2010.

[104] H. Huo, Y. Xu, C. C. Bilen, and H. Zhang. Coexistence issues of 2.4ghz sensor networks with other rf devices at home. In *Proceedings of the* 3^{rd} *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pages 200–205, June 2009.

[105] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso: A multi-channel approach to mitigate external interference. In *Proceedings of the* 8^{th} *IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, pages 449–457, June 2011.

[106] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the* 5^{th} *International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 1–12, Dec. 2009.

[107] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *Proceedings of the* 13^{th} *ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 409–420, Aug. 2007.

[108] J. Jeong. *Wireless Sensor Networking for Intelligent Transportation Systems.* PhD thesis, University of Minnesota, MN, USA, Dec. 2009.

[109] Y. Jeong, J. Kim, and S.-J. Han. Interference mitigation in wireless sensor networks using dual heterogeneous radios. *Wireless Networks*, 17(7):1699–1713, Oct. 2011.

[110] A. Jiménez-González, J. R. M. de Dios, and A. Ollero. An integrated testbed for heterogeneous mobile robots and other cooperating objects. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3327–3332, Oct. 2010.

[111] A. Jiménez-González, J. R. M. de Dios, and A. Ollero. Towards an open testbed for the cooperation of robots and wireless sensor networks. In *Proceedings of the* 10^{th} *Conference on Mobile Robots and Competitions (Robotica)*, Mar. 2010.

[112] D. Johnson, T. Stack, R. Fish, D. M. Flickingery, L. Stoller, R. Ricci, and J. Lepreau. Mobile Emulab: A robotic wireless and sensor network testbed. In *Proceedings of the* 25th *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–12, Apr. 2006.

[113] X. Ju, H. Zhang, and D. Sakamuri. NetEye: A user-centered wireless sensor network testbed for high-fidelity, robust experimentation. *International Journal of Communication Systems*, 25(9):1213–1229, Sept. 2012.

[114] A. Kamerman and N. Erkocevic. Microwave oven interference on wireless LANs operating in the 2.4 ghz ISM band. In *Proceedings of the* 8th *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIRMC)*, volume 3, pages 1221–1227, Sept. 1997.

[115] M. S. Kang, J. W. Chong, H. Hyun, S. M. Kim, B. H. Jung, and D. K. Sung. Adaptive interference-aware multi-channel clustering algorithm in a zigbee network in the presence of WLAN interference. In *Proceedings of the* 2nd *International Symposium on Wireless Pervasive Computing (ISWPC)*, Feb. 2007.

[116] A. Kanzaki, T. Hara, Y. Ishi, N. Wakamiya, and S. Shimojo. X-Sensor: A sensor network testbed integrating multiple networks. In *Proceedings of the* 3rd *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pages 1082–1087, Mar. 2009.

[117] C. Keppitiyagama, N. Tsiftes, C. A. Boano, and T. Voigt. Poster abstract: Temperature hints for sensornet routing. In *Proceedings of the* 11th *ACM Conference on Embedded Networked Sensor Systems (SenSys), poster session*, pages 25:1–25:2. ACM, Nov. 2013.

[118] B. Kerkez, T. Watteyne, M. Magliocco, S. Glaser, and K. Pister. Feasibility analysis of controller design for adaptive channel hopping. In *Proceedings of the* 4th *International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS*, pages 76:1–76:6, Oct. 2009.

[119] S. Kim, S. Pakzad, D. E. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the* 6th *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 254–263, Apr. 2007.

[120] 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 *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 53–63, Apr. 2008.

[121] Y. J. Kim, R. G. Evans, and W. M. Iversen. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1379–1387, July 2008.

[122] J. Ko, J. H. Lim, Y. Chen, R. Musvaloiu-E., A. Terzis, G. M. Masson, T. Gao, W. Destler, L. Selavo, and R. P. Dutton. MEDiSN: Medical emergency detection

in sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 10(1):1–29, Aug. 2010.

[123] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea. In *Proceedings of the* 3^{rd} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 64–75, Nov. 2005.

[124] B. Kusy, C. Richter, W. Hu, M. Afanasyev, R. Jurdak, M. Brünig, D. Abbott, C. Huynh, and D. Ostry. Radio diversity for reliable communication in WSNs. In *Proceedings of the* 10^{th} *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 270–281, Apr. 2011.

[125] O. Landsiedel, F. Ferrari, and M. Zimmerling. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *Proceedings of the* 11^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–14, Nov. 2013.

[126] K. G. Langendoen, A. Baggio, and O. W. Visser. Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In *Proceedings of the* 14^{th} *International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, Apr. 2006.

[127] S.-Y. Lau, T.-H. Lin, T.-Y. Huang, I.-H. Ng, and P. Huang. A measurement study of zigbee-based indoor localization systems under rf interference. In *Proceedings of the* 4^{th} *ACM International Workshop on Experimental Evaluation and Characterization (WINTECH)*, pages 35–42, Sept. 2009.

[128] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The trickle algorithm. Technical report, IETF, Mar. 2011.

[129] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. L. Hill, M. Welsh, E. Brewer, and D. E. Culler. TinyOS: An operating system for sensor networks. In *Ambient Intelligence*. Springer Berlin Heidelberg, 2005.

[130] P. Levis, N. Patel, D. E. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the* 1^{st} *Conference on Symposium on Networked Systems Design and Implementation (NSDI)*, pages 15–28, Mar. 2004.

[131] C.-J. M. Liang. *Interference Characterization and Mitigation in Large-Scale Wireless Sensor Networks*. PhD thesis, John Hopkins University, Baltimore, MD, USA, Jan. 2011.

[132] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the* 8^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 309–322, Nov. 2010.

[133] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the* 12*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 153–166, Apr. 2013.

[134] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proceedings of the* 4*th* *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 223–236, Nov. 2006.

[135] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. A. Stankovic, and T. He. Towards stable network performance in wireless sensor networks. In *Proceedings of the* 30*th* *IEEE International Real-Time Systems Symposium (RTSS)*, pages 227–237, Dec. 2009.

[136] T. Liu and A. Cerpa. Foresee (4c): Wireless link prediction using link features. In *Proceedings of the* 10*th* *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 294–305, Apr. 2011.

[137] K. Lorincz, B. rong Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: A wearable sensor network platform for high-fidelity motion analysis. In *Proceedings of the* 7*th* *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 183–196, Nov. 2009.

[138] L. Luo, T. He, G. Zhou, L. Gu, T. F. Abdelzaher, and J. A. Stankovic. Asynchronous events in wireless sensor networks with EnviroLog. In *Proceedings of the* 25*th* *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–14, Apr. 2006.

[139] A. Mainwaring, D. E. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the* 1*st* *International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 88–97, Sept. 2002.

[140] R. Marfievici, A. L. Murphy, G. P. Picco, F. Ossi, and F. Cagnacci. How environmental factors impact outdoor wireless sensor networks: A case study. In *Proceedings of the* 10*th* *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, pages 565–573, Sept. 2013.

[141] K. Martinez, R. Ong, and J. K. Hart. GLACSWEB: A sensor network for hostile environments. In *Proceedings of the* 1*st* *International Conference on Sensor and Ad-Hoc Communications and Networks (SECON)*, pages 81–87, Oct. 2004.

[142] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali. Routing without routes: the backpressure collection protocol. In *Proceedings of the* 9*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 279–290, Apr. 2010.

[143] L. Mottola, G. P. Picco, M. Ceriotti, S. Gună, and A. L. Murphy. Not all wireless sensor networks are created equal: A comparative study on tunnels. *ACM Transactions on Sensor Networks (TOSN)*, 7(2):15:1–33, Sept. 2010.

[144] R. Musaloiu-E., C.-J. M. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *Proceedings of the* 7^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 421–432, Apr. 2008.

[145] R. Musaloiu-E. and A. Terzis. Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. *International Journal of Sensor Networks (IJSNet)*, 3(1):43–54, Dec. 2007.

[146] K. Na, Y. Kim, and H. Cha. Acoustic sensor network-based parking lot surveillance system. In *Proceedings of the* 6^{th} *European Conference on Wireless Sensor Networks (EWSN)*, pages 247–262, Feb. 2009.

[147] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-Power Listening Goes Multi-Channel. In *Proceedings of the* 10^{th} *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2014.

[148] M. Nilsson. Directional antennas for wireless sensor networks. In *Proceedings of the* 9^{th} *Scandinavian Workshop on Wireless Adhoc Network (Adhoc)*, May 2009.

[149] C. Noda, S. Prabh, M. Alves, C. A. Boano, and T. Voigt. Quantifying the channel quality for interference-aware wireless sensor networks. *ACM SIGBED Review*, 8(4):43–48, Dec. 2011.

[150] F. J. Oppermann, C. A. Boano, and K. Römer. A decade of wireless sensing applications: Survey and taxonomy. In *The Art of Wireless Sensor Networks*, volume 1 of *Signals and Communication Technology*, chapter 2, pages 11–50. Springer Berlin Heidelberg, 2014.

[151] F. J. Oppermann, C. A. Boano, K. Römer, and M. Zimmerling. Automatic configuration of controlled interference experiments in sensornet testbeds. In *Proceedings of the* 12^{th} *ACM International Conference on Embedded Networked Sensor Systems (SenSys), poster session*. ACM, Nov. 2014.

[152] J. Ortiz and D. Culler. Multichannel reliability assessment in real world WSNs. In *Proceedings of the* 9^{th} *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 162–173, Apr. 2010.

[153] F. Österlind. *Improving Low-Power Wireless Protocols with Timing-Accurate Simulation*. PhD thesis, Uppsala University, Uppsala, Sweden, Nov. 2011.

[154] F. Österlind, A. Dunkels, T. Voigt, N. Tsiftes, J. Eriksson, and N. Finne. Sensornet check-pointing: Enabling repeatability in testbeds and realism in simulators. In *Proceedings of the* 6^{th} *European Conference on Wireless Sensor Networks (EWSN)*, pages 343–357, Feb. 2009.

[155] F. Österlind, L. Mottola, T. Voigt, N. Tsiftes, and A. Dunkels. Strawman: Resolving collisions in bursty low-power wireless networks. In *Proceedings of the* 11^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 161–172, Apr. 2012.

[156] E. Öström, L. Mottola, M. Nilsson, and T. Voigt. Smart antennas made practical: the SPIDA way. In *Proceedings of the* 9th *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), demo session*, pages 438–439, Apr. 2010.

[157] Özlem Durmaz Incel, P. Jansen, and S. Mullender. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Journal of Ad Hoc Networks*, 9:73–94, Jan. 2011.

[158] S. N. Pakzad. The GINSENG system for wireless monitoring and control: Design and deployment experiences. *ACM Transactions on Sensor Networks (TOSN)*, 10(1):4:1–4:40, Nov. 2013.

[159] C. Park, K. Lahiri, and A. Raghunathan. Battery discharge characteristics of wireless sensor nodes: An experimental analysis. In *Proceedings of the* 2nd *IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 430–440, Sept. 2005.

[160] F. Penna, C. Pastrone, M. Spirito, and R. Garello. Measurement-based analysis of spectrum sensing in adaptive WSNs under wi-fi and bluetooth interference. In *Proceedings of the* 69th *IEEE Vehicular Technology Conference (VTC)*, pages 1–5, Apr. 2009.

[161] M. Petrova, L. Wu, P. Mähönen, and J. Riihijärvi. Interference measurements on performance degradation between colocated ieee 802.11g/n and ieee 802.15.4 networks. In *Proceedings of the* 6th *International Conference on Networking (ICN)*, pages 93–98, Apr. 2007.

[162] K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *Proceedings of the IASTED International Symposium on Distributed Sensor Networks (DSN)*, pages 391–398, Nov. 2008.

[163] K. S. Pister. Tracking vehicles with a UAV-delivered sensor network. Technical report, UC Berkeley and MLB, Mar. 2001.

[164] J. Polastre, J. L. Hill, and D. E. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the* 2nd *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 95–107, Nov. 2004.

[165] J. Polastre, R. Szewczyk, and D. E. Culler. Telos: Enabling ultra-low power wireless research. In *Proceedings of the* 4th *International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 364–369, Apr. 2005.

[166] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of wireless sensor networks for habitat monitoring. In *Wireless Sensor Networks*, pages 399–423. Kluwer Academic Publishers, 2004.

[167] W.-B. Pöttner, S. Schildt, D. Meyer, and L. Wolf. Piggy-backing link quality measurements to IEEE 802.15.4 ACKs. In *Proceedings of the* 8th *International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, pages 807–812, Oct. 2011.

[168] D. Puccinelli and S. Giordano. ViMobiO: Virtual mobility overlay for static sensor network testbeds. In *Proceedings of the* 4^{th} *IEEE International Workshop on Advanced Experimental Activities on Wireless Networks and Systems (EXPonWireless)*, pages 1–6, June 2009.

[169] Y. Qin, Z. He, and T. Voigt. Towards accurate and agile link quality estimation in wireless sensor networks. In *Proceedings of the* 10^{th} *IFIP Annual Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net)*, pages 179–185, June 2011.

[170] M. H. Rehmani, T. Alves, S. Lohier, A. Rachedi, and B. Poussot. Towards intelligent antenna selection in IEEE 802.15.4 wireless sensor networks. In *Proceedings of the* 13^{th} *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 245–246, June 2012.

[171] O. Rensfelt, F. Hermans, L.-A. Larzon, and P. Gunningberg. Sensei-UU: a relocatable sensor network testbed. In *Proceedings of the* 5^{th} *ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*, pages 63–70, Sept. 2010.

[172] M. Ringwald. *Reducing Uncertainty in Wireless Sensor Networks.* PhD thesis, Eidgenössische Technische Hochschule (ETH), Zürich, Switzerland, 2009.

[173] C. Rohner, L. M. Feeney, and P. Gunningberg. Evaluating battery models in wireless sensor networks. In *Proceedings of the* 11^{th} *International Conference on Wired/Wireless Internet Communications (WWIC)*, pages 29–42, June 2013.

[174] C. B. D. Rosiers, G. Chelius, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, and T. Noël. SensLAB: Very large scale open wireless sensor network testbed. In *Proceedings of the* 7^{th} *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, pages 239–254, Apr. 2011.

[175] W. Roush. 10 emerging technologies that will change the world. *MIT Technology Review*, 106(1):33–49, Feb. 2003.

[176] A. Rowe, V. Gupta, and R. Rajkumar. Low-power clock synchronization using electromagnetic energy radiating from AC power lines. In *Proceedings of the* 7^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 211–224, Nov. 2009.

[177] P. L. Ryan. Radio frequency propagation differences through various transmissive materials. Master's thesis, University of North Texas, Denton, TX, USA, Dec. 2002.

[178] M. Salajegheh, H. Soroush, and A. Kalis. HyMAC: Hybrid TDMA/FDMA medium access control protocol for wireless sensor networks. In *Proceedings of the* 18^{th} *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–5, Sept. 2007.

[179] J. Sallai, Ákos Lédeczi, and P. Völgyesi. Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes. In *Proceedings of the* 9^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 96–107, Nov. 2011.

[180] San Francisco Municipal Transportation Agency. SFpark: Putting theory into practice, Aug. 2011.

[181] A. Sanchez, I. Moerman, S. Bouckaert, D. Willkomm, J.-H. Hauer, N. Michailow, G. Fettweis, L. Dasilva, J. Tallon, and S. Pollin. Testbed federation: An approach for experimentation-driven research in cognitive radios and cognitive networking. In *Proceedings of the of the 20th Future Network and Mobile Summit*, pages 1–9, June 2011.

[182] T. Schmid. *Time in Wireless Embedded Systems*. PhD thesis, University of California, Los Angeles, CA, USA, 2009.

[183] T. Schmid, Z. Charbiwala, R. Shea, and M. B. Srivastava. Temperature compensated time synchronization. *IEEE Embedded Systems Letters*, 1(2):37–41, Aug. 2009.

[184] R. Sen, A. Maurya, B. Raman, R. Mehta, R. Kalyanaraman, N. Vankadhara, S. Roy, and P. Sharma. Kyun Queue: A sensor network system to monitor road traffic queues. In *Proceedings of the* 10^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 127–140, Nov. 2012.

[185] M. Sha, G. Hackmann, and C. Lu. ARCH: Practical channel hopping for reliable home-area sensor networks. In *Proceedings of the* 17^{th} *International Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 305–315, Apr. 2011.

[186] M. Sha, G. Hackmann, and C. Lu. Multi-channel reliability and spectrum usage in real homes: Empirical studies for home-area sensor networks. In *Proceedings of the* 19^{th} *International Workshop on Quality of Service (IWQoS)*, pages 1–9, June 2011.

[187] M. Sha, G. Hackmann, and C. Lu. Energy-efficient low power listening for wireless sensor networks in noisy environments. In *Proceedings of the* 12^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 277–288, Apr. 2013.

[188] K. Shuaib, M. Boulmalf, F. Sallabi, and A. Lakas. Co-existence of zigbee and WLAN: a performance study. In *Proceedings of the* 5^{th} *IEEE International Wireless Telecommunications Symposium (WTS)*, pages 1–6, Apr. 2006.

[189] A. Sikora and V. F. Groza. Coexistence of IEEE 802.15.4 with other systems in the 2.4 ghz-ISM-band. In *Proceedings of the IEEE Conference on Instrumentation and Measurement Technology (IMTC)*, pages 1786–1791, May 2005.

[190] J. Slipp, C. Ma, N. Polu, J. Nicholson, M. Murillo, and S. Hussain. WINTeR: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. In *Proceedings of the* 6^{th} *IEEE International Conference*

on Communication Networks and Services Research (CNSR), pages 422–431, May 2008.

[191] H.-S. W. So, J. Walrand, and J. Mo. McMAC: A parallel rendezvous multi-channel MAC protocol. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 334–339, Mar. 2007.

[192] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the* 4^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 237–250, Nov. 2006.

[193] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt. WirelessHART: Applying wireless technology in real-time industrial process control. In *Proceedings of the* 14^{th} *IEEE International Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 377–386, Apr. 2008.

[194] L. Stabellini and J. Zander. Energy-efficient detection of intermittent interference in wireless sensor networks. *International Journal of Sensor Networks (IJSNET)*, 8(1):27–40, July 2010.

[195] L. P. Steyn and G. P. Hancke. A survey of wireless sensor network testbeds. In *Proceedings of the* 10^{th} *IEEE AFRICON Conference*, pages 1–6, Sept. 2011.

[196] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline. PIPENET: a wireless sensor network for pipeline monitoring. In *Proceedings of the* 6^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 264–273, Apr. 2007.

[197] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *Proceedings of the* 3^{rd} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 13–26, Nov. 2005.

[198] J. Sun and R. C. Oliver. An experimental evaluation of temporal characteristics of communication links in outdoor sensor networks. In *Proceedings of the* 2^{nd} *Workshop on Real-World Wireless Sensor Networks (REALWSN)*, pages 73–77, June 2006.

[199] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition wireless sensor networks. *Wireless Sensor Networks*, 2920:307–322, 2004.

[200] T. M. Taher, M. J. Misurac, J. L. LoCicero, and D. R. Ucci. Microwave oven signal modeling. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1235–1238, Apr. 2008.

[201] M. Tancreti, M. S. Hossain, S. Bagchi, and V. Raghunathan. AVEKSHA: A hardware-software approach for non-intrusive tracing and profiling of wireless embedded systems. In *Proceedings of the* 9^{th} *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 288–301, Nov. 2011.

[202] 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* 12*th* *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 23:1–23:11, May 2011.

[203] J. Tateson, C. Roadknight, A. Gonzalez, T. Khan, S. Fitz, I. Henning, N. Boyd, C. Vincent, and I. Marshall. Real world issues in deploying a wireless sensor network for oceanography. In *Proceedings of the* 1*st* *Workshop on Real-world Wireless Sensor Networks (REALWSN)*, pages 23:1–23:11, June 2005.

[204] S. Technologies. T301 testbed. `http://www.sownet.nl/download/2009-01-26testbed301web.pdf`, Jan. 2009.

[205] Texas Instruments. *CC2520 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee RF Transceiver*, revision swrs068 edition, dec 2007.

[206] Texas Instruments. *CC2420 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, revision swrs041c edition, Feb. 2013.

[207] J. Thelen, D. Goense, and K. Langendoen. Radio wave propagation in potato fields. In *Proceedings of the* 1*st* *Workshop on Wireless Network Measurement (WiNMee)*, Apr. 2005.

[208] N. Thepvilojanapong, T. Ono, and Y. Tobe. A deployment of fine-grained sensor network and empirical analysis of urban temperature. *Sensors*, 10:2217–2241, Mar. 2010.

[209] G. Tolle, J. Polastre, R. Szewczyk, D. E. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. In *Proceedings of the* 3*rd* *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 51–63, Nov. 2005.

[210] N. Tsiftes, A. Dunkels, Z. He, and T. Voigt. Enabling large-scale storage in sensor networks with the Coffee file system. In *Proceedings of the* 8*th* *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 349–360, Apr. 2009.

[211] N. Tsiftes, J. Eriksson, and A. Dunkels. Low-power wireless IPv6 routing with ContikiRPL. In *Proceedings of the* 9*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 406–407, Apr. 2010.

[212] N. Tsiftes, T. Voigt, F. Aslam, I. Protonotarios, M. A. Zúñiga, K. Langendoen, C. A. Boano, F. J. Oppermann, K. Römer, M. Baunach, J. Brown, U. Roedig, P. M. Montero, R. S. Hernández, and J. C. P. Marius Montón. D-4.3 - first integrated prototype and experiment. Technical report, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, May 2014.

[213] T. Voigt, L. Mottola, and K. Hewage. Understanding link dynamics in wireless sensor networks with dynamically steerable directional antennas. In *Proceedings of the* 10*th*

European Conference on Wireless Sensor Networks (EWSN), pages 115–130, Feb. 2013.

[214] T. Voigt and F. Österlind. CoReDac: Collision-free command-response data collection. In *Proceedings of the* 13^{th} *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept. 2008.

[215] T. Voigt, F. Österlind, and A. Dunkels. Improving sensor network robustness with multi-channel convergecast. In *Proceedings of the* 2^{nd} *ERCIM Workshop on e-Mobility*, May 2008.

[216] M. Vollmer. Physics of the microwave oven. *Physics Education*, 39:74–81, Jan. 2004.

[217] L. Wanner, C. Apte, R. Balani, P. Gupta, and M. Srivastava. Hardware variability-aware duty cycling for embedded sensors. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(6):1000–1012, June 2013.

[218] T. Watteyne, S. Lanzisera, A. Mehta, and K. S. Pister. Mitigating multipath fading through channel hopping in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1–5, May 2010.

[219] T. Watteyne, A. Mehta, and K. Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the 6th International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 116–123, Oct. 2009.

[220] H. Wennerström, F. Hermans, O. Rensfelt, C. Rohner, and L.-A. Nordén. A long-term study of correlations between meteorological conditions and 802.15.4 link performance. In *Proceedings of the* 10^{th} *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 221–229, June 2013.

[221] H. Wennerström, L. McNamara, C. Rohner, and L.-A. Nordén. Transmission errors in a sensor network at the edge of the world. In *Proceedings of the* 5^{th} *Extreme Conference on Communication (ExtremeCom)*, pages 19–24, Aug. 2013.

[222] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the* 7^{th} *International Symposium on Operating Systems Design and Implementation (OSDI)*, pages 381–396, Nov. 2006.

[223] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, Mar. 2006.

[224] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a wireless sensor network testbed. In *Proceedings of the* 4^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 483–488, Apr. 2005.

[225] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 routing protocol for low-power and lossy networks. Technical report, IETF, Mar. 2012.

[226] M. Woehrle, M. Bor, and K. Langendoen. 868 mhz: a noiseless environment, but no free lunch for protocol design. In *Proceedings of the* 9*th* *International Conference on Networked Sensing Systems (INSS)*, pages 1–8, June 2012.

[227] C. Won, J.-H. Youn, H. Ali, H. Sharif, and J. Deogun. Adaptive radio channel allocation for supporting coexistence of 802.15.4 and 802.11b. In *Proceedings of the* 62*nd* *IEEE Vehicular Technology Conference (VTC)*, pages 2522–2526, Sept. 2005.

[228] Y. Wu, M. Keally, G. Zhou, and W. Mao. Traffic-aware channel assignment in wireless sensor networks. In *Proceedings of the* 4*th* *International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, pages 479–488, Aug. 2009.

[229] Y. Wu, J. A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Proceedings of the* 27*th* *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1193–1201, Apr. 2008.

[230] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the* 2*nd* *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 13–24, Nov. 2004.

[231] R. Xu, G. Shi, J. Luo, Z. Zhao, and Y. Shu. MuZi: Multi-channel zigbee networks for avoiding wifi interference. In *Proceedings of the* 4*th* *International Conference on Cyber, Physical and Social Computing (CPSCOM)*, pages 323–329, June 2011.

[232] W. Xu, W. Trappe, and Y. Zhang. Defending wireless sensor networks from radio interference through channel adaptation. *ACM Transactions on Sensor Networks (TOSN)*, 4, Aug. 2008.

[233] S. Yamashita, T. Shimura, K. Aiki, K. Ara, Y. Ogata, I. Shimokawa, T. Tanaka, H. Kuriyama, K. Shimada, and K. Yano. A 15x15 mm, 1 /spl mu/a, reliable sensor-net module: Enabling application-specific nodes. In *Proceedings of the* 5*th* *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 383–390, Apr. 2006.

[234] D. Yang, Y. Xu, and M. Gidlund. Wireless coexistence between ieee 802.11- and ieee 802.15.4-based networks: A survey. *International Journal of Distributed Sensor Networks (IJDSN)*, 2011(912152), Apr. 2011.

[235] Z. Yang, L. Cai, Y. Liu, and J. Pan. Environment-aware clock skew estimation and synchronization for wireless sensor networks. In *Proceedings of the* 31*st* *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1017–1025, Mar. 2012.

[236] S.-U. Yoon, R. Murawski, E. Ekici, S. Park, and Z. H. Mir. Adaptive channel hopping for interference robust wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 432–439, May 2010.

[237] W. Yuan, J.-P. M. Linnartz, and I. G. Niemegeers. Adaptive CCA for IEEE 802.15.4 wireless sensor networks to mitigate interference. In *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC)*, pages 1–5, Apr. 2010.

[238] Z. Zhong and T. He. Achieving range-free localization beyond connectivity. In *Proceedings of the* 7^{th} *International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 281–294, Nov. 2009.

[239] G. Zhou, L. Lu, S. Krishnamurthy, M. Keally, and Z. Ren. SAS: Self-adaptive spectrum management for wireless sensor networks. In *Proceedings of the* 18^{th} *Internatonal Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, Aug. 2009.

[240] G. Zhou, J. A. Stankovic, and S. H. Son. Crowded spectrum in wireless sensor networks. In *Proceedings of the* 3^{rd} *Workshop on Embedded Networked Sensors (EmNets)*, May 2006.

[241] R. Zhou and G. Xing. Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks. In *Proceedings of the* 12^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 141–152, Apr. 2013.

[242] M. A. Zúñiga, F. Aslam, I. Protonotarios, K. Langendoen, C. A. Boano, K. Römer, J. Brown, U. Roedig, N. Tsiftes, and T. Voigt. D-2.1 - report on optimized and newly designed protocols. Technical report, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, May 2014.

[243] M. A. Zúñiga, C. A. Boano, J. Brown, C. Keppitiyagama, F. J. Oppermann, P. Alcock, N. Tsiftes, U. Roedig, K. Römer, T. Voigt, and K. Langendoen. D-1.1 - report on environmental and platform models. Technical report, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, June 2013.

[244] M. A. Zúñiga, I. Irzynska, J.-H. Hauer, T. Voigt, C. A. Boano, and K. Römer. Link quality ranking: Getting the best out of unreliable links. In *Proceedings of the* 7^{th} *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 1–8. IEEE, June 2011.

[245] M. A. Zúñiga and B. Krishnamachari. Analyzing the transitional region in low-power wireless links. In *Proceedings of the* 1^{st} *IEEE International Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 517–526, Oct. 2004.

[246] M. A. Zúñiga and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Transactions on Sensor Networks (TOSN)*, 3(2), June 2007.