Reliable and Available Low-Power Wireless Mesh Networks

Georgios Papadopoulos

Reliable and Available Low-Power Wireless Mesh Networks

Georgios Z. PAPADOPOULOS

Habilitation à Diriger des Recherches

Université de Rennes 1

Presented on 22 November 2021.

Jury:

Prof. Fabrice VALOIS	INSA Lyon, France	President
Prof. Carla Fabiana CHIASSERINI	Polytechnic University of Turin, Italy	Reviewer
Dr. Nathalie Mitton	Inria Lille-Nord Europe, France	Reviewer
Prof. Xavier VILAJOSANA GUILLEN	Open University of Catalonia, Spain	Reviewer
Dr. Marcelo DIAS DE AMORIM	Sorbonne University, France	Examiner
Prof. Martin HEUSSE	Ensimag / Grenoble INP, France	Examiner
Prof. Cesar VIHO	University of Rennes 1, France	Examiner

This version of the manuscript was compiled on June 26, 2023.

Αφιερωμένο στην Ευγενία και στην κόρη μου Αριάδνη Λάρα.

Acknowledgements

The current "Habilitation à diriger des recherches (HDR)" is the result of six intensive years of exciting work at the IMT Atlantique, campus of Rennes, in France.

There are two persons I would like to cite, and who have had a profound impact on my research, Nicolas Montavont and Pascal Thubert. Since 2016, you have been providing me with continuous guidance and support, and I would like to thank you for that!

This habilitation would not have been possible without the hard work of (in chronological order) Vasilios Kotsiou, Tadanori Matsui, Julian Martin Del Fiore, Remous-Aris Koutsiamanis, Tomas Lagos Jenschke, Renzo Efraín Navas, Guillaume Le Gall, Alexandre Marquet, Chenyang Ji, Eduardo Inglés Sánchez, Sergio Aguilar Romero, David Hauweele, Ana Czarnitzki Estrin, Dimitris Sourailidis, Fabián Antonio Rincon Vija, and Amaury Bruniaux! You should know that I have been privileged to serve as advisor and to collaborate with you all!

One should never forget his roots; I therefore would like to thank (in chronological order) Periklis Chatzimisios, Xavier Costa-Pérez, Andres Garcia-Saavedra, Pablo Serrano, Antoine Gallais, Thomas Noel, George Oikonomou, and Theo Tryfonas.

I would also like to thank the members of my jury, and in particular the reviewers. It is an honor for me to be able to present my research work to you.

Contents

List of Abbreviations

3GPP	3rd Generation Partnership Project.
5G	Fifth Generation Mobile Networks.
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks.
6P	6top Protocol.
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e.
6top	6TiSCH Operation Sublayer.
ACK	acknowledgment.
ADC	Analog-to-Digital Converter.
AMABO	Adaptive MABO.
AP	Alternative Parent.
ARQ	Automatic Repeat reQuest.
ASN	Absolute Sequence Number.
BMS	Battery Management System.
CA	Common Ancestor.
CBR	Constant Bit Rate.
CoAP	Constrained Application Protocol.
CoRE	Constrained RESTful Environments.
CSU	Cell Sensor Unit.
DAG	Directed Acyclic Graph.
DAO	Destination Advertisement Object.
DetNet	Deterministic Networking.
DIO	DODAG Information Object.
DIS	DODAG Informational Solicitation.
DODAG	Destination Oriented Directed Acyclic Graph.
DSSS	Direct Sequence Spread Spectrum.
EB	Enhanced Beacon.
ETX	Expected Transmission Count.
EV	Electric Vehicle.
FDMA	Frequency Division Multiple Access.
FEC	Forward Error Correction.

HARQ	Hybrid ARQ.
HC	Hop Counting.
I-D	Internet Draft.
IAB	Integrated Access and Backhaul.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
IoT	Internet of Things.
IP	Internet Protocol.
IPv6	Internet Protocol version 6.
ISM	Industrial, Scientific and Medical.
LABeL	Link-based Adaptive BLacklisting.
LDSF	Low-latency Distributed Scheduling Function.
LLN	Low Power and Lossy Network.
LLSF	Low Latency Scheduling Function.
LP	Linear Programming.
LPWAN	Low-Power Wide Area Network.
LQI	Link Quality Indicator.
MABO-TSCH	Multi-hop And Blacklist-based Optimised TSCH.
MAC	Medium Access Control.
MC	Metric Container.
MCU	Master Control Unit.
MSF	6TiSCH Minimal Scheduling Function.
MTD	Moving Target Defense.
MTU	Maximum Transmission Unit.
NSA	Node State and Attribute.
ODeSe	On-Demand Selection.
OF	Objective Function.
OH	Overhearing.
PAREO	Packet Automatic Repeat reQuest, Replication and Elimination, and Overhearing.
PCE	Path Computation Element.
PDR	Packet Delivery Ratio.
PLC	Power Line Communication.
PP	Preferred Parent.
PS	Parent Set.
QoS	Quality of Service.
RAW	Reliable and Available Wireless.
RDC	Radio Duty Cycle.
RE	Replication and Elimination.
RLC	Radio Link Control.

ROLL	Routing Over Low power and Lossy networks.
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks.
RSSI	Received Signal Strength Indicator.
RTX	Retransmission.
SD	Simple Descent.
SDN	Software-Defined Networking.
SDR	Software-Defined Radio.
SF	Scheduling Function.
TDMA	Time Division Multiple Access.
TSCH	Time-Slotted Channel Hopping.
UCTP	University Course Timetabling Problem.
WG	Working Group.
WMEWMA	Window Mean Exponentially-Weighted Moving Average.
WSN	Wireless Sensor Network.

Avant-propos

I defended my PhD thesis six years ago in 2015. At that time, my research activities included mainly two topics: the *contention-based* Medium Access Control (MAC) and the radio characterisation. In the first topic, I worked on enhancing the contention-based Radio Duty Cycle (RDC) MAC protocols in Wireless Sensor Networks (WSNs) that are designed primarily for energy saving purposes. The second part was on investigating the role of simulators and testbeds in the research process cycle, and to identify the means to strengthen their complementarity. Moreover, the goal was to demonstrate the importance of repeatable experimental setups for reproducible performance evaluation results.

Today, the attention from the academic world regarding the first topic has been moved to other research areas. Thanks to the research questions that I had raised in my second topic, I realised the importance of *contention-free* and channel hopping MAC protocols. Therefore, I moved towards IEEE Std 802.15.4-2015 Time-Slotted Channel Hopping (TSCH). As a result, since 2016, I have been working on enhancing the TSCH protocol as well as the whole 6TiSCH stack, in order to provide reliable and available low-power wireless mesh networking. The outcome of these efforts are several successful PhD theses, many scientific articles, contributions at standardisation organisations such as the Internet Engineering Task Force (IETF), successful research projects and collaborations with industrial partners, and most importantly *some good friends*!

In brief, I have developed during the past six years a more solid expertise in (i) radio blacklisting algorithms, (ii) resource allocation schemes and scheduling functions, and (iii) multi-path routing protocols. I have applied this triple expertise in the area of low-power wireless mesh networks. In this manuscript, in order to keep a consistent story, I had to omit some of my research activities of which I am very proud. Some of these works are: (i) "Improving the resilience of the constrained Internet of Things (IoT)", the PhD thesis topic of Renzo E. Navas, (ii) "Software-Defined Radio (SDR) implementation of LoRa and demodulation techniques", the Postdoctoral work of Alexandre Marquet, and (iii) "Performance Evaluation of the fragmentation and compression protocols in Low-Power Wide Area Network (LPWAN)", the PhD visit topic of Sergio Aguilar Romero.

Through contributions in the academic and standardisation worlds, I hope to have had impact in the field of low-power wireless mesh networks. I hope you will have as much fun reading this manuscript as I had writing it.

Chapter 1

Introduction

1.1 Low-Power Wireless Mesh Networking

After many decades of research and development, wireless networks have evolved from the Ad Hoc Networks technologies to Low-Power Wireless Mesh Networks, in which "smart", uniquely identifiable and wirelessly connected objects (e.g., sensors, actuators) cooperatively construct a wireless multi-hop (and potentially large-scale) network of things, commonly known as the Internet of Things (IoT). These "things", objects, or constrained devices can communicate with each other or across existing network infrastructure such as the Internet. They, most of the times come with limited capacity in terms of memory storage, computational power and energy.

This evolution of wireless networks in conjunction with the low production cost of the constrained devices and their ease deployment, enabled the design of radically new applications that follow the modern concept of IoT. Indeed, low-power wireless mesh networks can be deployed nearly everywhere where sensing or actuation services are required. For instance, they have been massively integrated into smart cities [1], smart farming [2], smart grid [3], [4], health monitoring [5], [6], and industrial applications [7].

Different types of applications require different levels of quality of service. Indeed, a given network of objects must be carefully designed for its target application. Among these IoT-based applications, critical applications, and, in particular industrial applications have been the core of my research during the past six years. Therefore, I am dedicating a whole Chapter, i.e., the Positioning Chapter 2, to present the research challenges that underpin my contributions both to the academic community and to standardisation bodies such as the Internet Engineering Task Force (IETF).

What is our "*thing*"?

A low-power electronic device (see Fig. 1.1), also known as node or mote, that is size of a matchbox performs operations such as sensing, actuation, detecting or responding to analogical inputs of its physical environment. These sensed measurements could be the light, the temperature, the humidity, the motion, the sound, the pressure, the vibration, or any other environmental phenomena. Such a device therefore embeds an Analog-to-Digital Converter (ADC) that

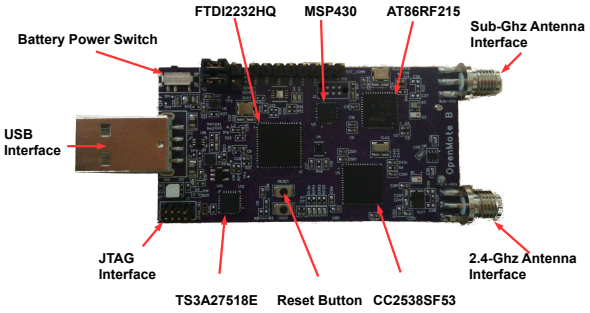


Fig. 1.1: The OpenMote B, a typical a low-power wireless node. *Taken from* [8].

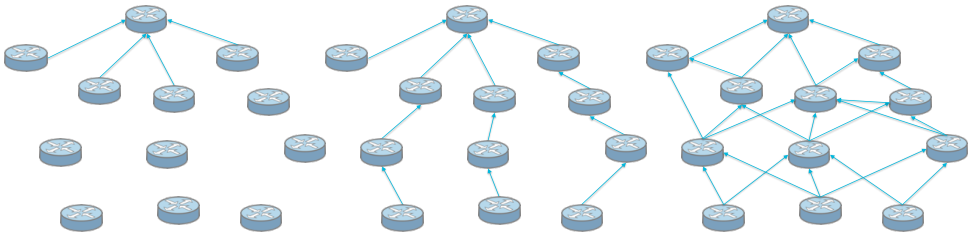
digitizes the continual analog signal produced by the sensors. A low-power node is made wireless once a wireless radio transceiver, e.g., AT86RF215 radio chip, is added. This radio chip enables a node to communicate with other nodes, and, thus it does not require wires to transmit or to receive sensor measurements. Moreover, the use of a wireless technology allows for flexible deployments and mobility-required applications. Furthermore, there is the micro-controller (e.g., CC2538) component that contains the flushed firmware, and it typically executes the network protocol stack and processes the measurements that are retrieved by the sensors. Other main components include an external memory and a power source which is typically powered by a battery.

The Potential Topologies

The low-power wireless mesh networks, just as its counterpart the Internet, is a structured network where constrained devices are organized in a given hierarchy. This network arrangement is called a topology, which indicates how the constrained devices are inter-connected. More specifically, low-power wireless mesh networks consist of sensors and actuators that operate in a given field. These constrained sensor and actuator nodes are able to communicate among each other and possibly also with external devices on the Internet. A very intuitive network arrangement is a set of nodes that collect environment measurements and send these measurements to a gateway, often called a border router. A gateway typically comes with multiple network interfaces, the wireless interface to communicate within the low-power wireless (mesh) network, and the wired interface to connect the low-power wireless (mesh) network to either a local computer network or to the "outside" world, the Internet. Therefore, a gateway is typically mains powered in order to be able to run more powerful software than the sensor or actuator nodes.

The following are the most popular network topologies that can be formed to enable the nodes to connect to the gateway, which are depicted in Fig. 1.2:

- star topology: where the nodes are within the radio propagation of the gateway, and, thus they can directly communicate with the gateway. The



(a) A star topology. (b) A tree topology. (c) A mesh topology.

Fig. 1.2: Possible logical topologies for low-power and lossy networks.

main advantage in this topology is its simplicity, meaning that the nodes turn ON their radio when they have something to transmit, which makes them very energy efficient. However, if the nodes are out of the propagation range of the gateway, they cannot participate in the network.

- tree topology: where some of the nodes might be not in the radio propagation of the gateway, thus, this approach let some of the nodes to operate as relays for others by forming a multi-hop tree topology that is rooted in the gateway. Even though the multi-hop tree topology overcomes the issue of a star topology, however, if one of the relaying nodes crashes or the link quality drops, all of its descendants in the network are disconnected.

- mesh topology: it is an extension of the tree topology to which one has added redundant paths. Each node in the network has at least two neighbors to transmit the data packet to. As a result, mesh topology overcomes the issue of a tree topology, thus even if some of the nodes go OFF, it will not impact the multi-hop networking as well as the traffic flow.

Mesh-based topologies are the most suitable and popular for industrial applications [9], [10], [11], [12]. Therefore, both from academic and industrial societies a lot of focus has been given on efficiently designing a mesh network while guaranteeing the envisioned trade-off level among the four elements: *i*) the network capacity (the data traffic the nodes can generate), *ii*) the end-to-end network latency, *iii*) the end-to-end network reliability, and *iv*) the energy consumption. A typical trade-off scenario for industrial applications is to target high network reliability and bounded latency at the cost of network capacity and energy consumption [13], [14], [15], [16], [17], [18].

In the following section, I present the networking software (i.e., the protocol stack) that runs on the constrained devices.

1.2 The Protocol Stack

In 2016 the IEEE Std 802.15.4-2015 standard [20] was published, and among the Medium Access Control (MAC) protocols defined in this standard, Time-Slotted Channel Hopping (TSCH) is the protocol to offer a certain level of quality of service. IEEE Std 802.15.4-TSCH is able to cope with the external interference and multi-path fading effect, which are the dominant causes of the radio link unreliability [21].

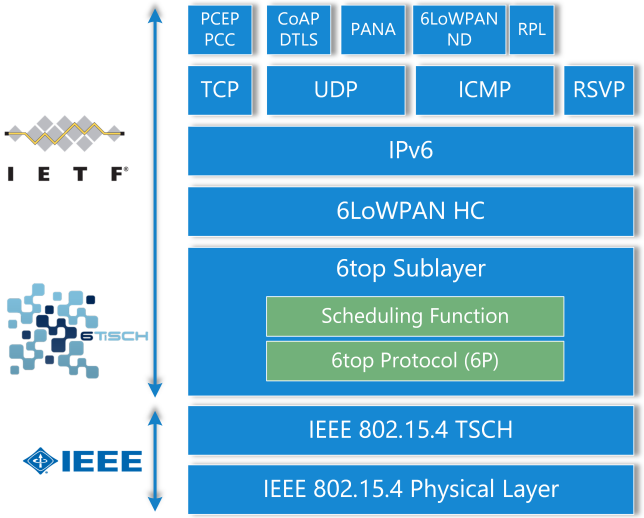


Fig. 1.3: Low Power and Lossy Network (LLN) protocol stack. *Taken from* [19].

At the IETF, a number of Working Groups (WGs) have been established to standardize a set of protocols for various layers of the LLN protocol stack compliant to IEEE Std 802.15.4-2015 radios, aiming to provide Internet Protocol version 6 (IPv6) connectivity to resource-constrained devices. Constrained RESTful Environments (CoRE) WG [22] defined the web transfer protocol, Constrained Application Protocol (CoAP) [23]. Routing Over Low power and Lossy networks (ROLL) WG [24] specified the routing protocol, IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [25]. IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) WG [26] introduced an adaptation layer by defining compression, fragmentation, reassembling and forwarding mechanisms for IPv6 datagrams that do not fit in the Maximum Transmission Unit (MTU) of 127*bytes* [27]. Finally, IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) WG [28] focuses on enabling IPv6 over the IEEE Std 802.15.4-2015 TSCH standard [20].

All these protocols form the LLN stack, see Fig. 1.3. Note that the first low layers are standardized at the Institute of Electrical and Electronics Engineers (IEEE), while the upper layers at the IETF.

In this section, the majority of the layers along with their key-protocols will be presented, with special focus given to IEEE Std 802.15.4-2015 TSCH, 6TiSCH, Scheduling Functions (SFs), and RPL, since these layers have been the base of my research during the past 6 years to provide end-to-end reliable networking and bounded latency in low-power mesh wireless networks.

IEEE Std 802.15.4-2015 TSCH

Under TSCH, communication among the nodes is orchestrated by a schedule, Fig. 1.4. At its core, TSCH uses a combination of the Time Division Multiple Access (TDMA) for the time dimension and the Frequency Division Multiple Access (FDMA) for the frequency dimension.

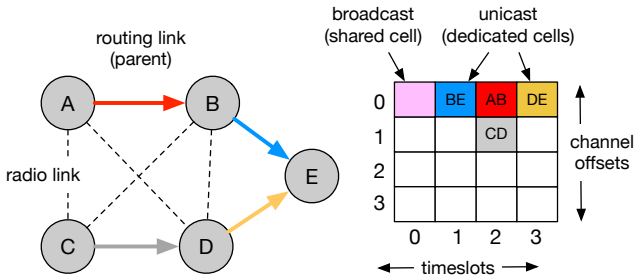


Fig. 1.4: TSCH schedule for a 5 nodes topology. *Taken from* [29].

The continuous time is divided into timeslots of equal length (typically $10ms$ long), sufficient enough for a node to transmit a frame and to receive an acknowledgement. The employed schedule indicates, for each timeslot whether the node has to stay "awake" (i.e., keep its radio ON) in order to transmit or receive a frame, or to "sleep" (turn its radio OFF) to save energy. A set of timeslots constructs a slotframe that repeats perpetually, and according to the standard it consists of 101 timeslots but it is configurable. The timeslots are identified by an Absolute Sequence Number (ASN) counter that increments as time elapses; the ASN actually counts the number of timeslots since the establishment of the TSCH network. All the connected nodes in the network are aware of the current ASN value.

The available frequency range into 16 non-overlapping physical radio channels operating at $2.4GHz$, where each radio channel has a bandwidth of $2MHz$ and a channel separation of $5MHz$.

Furthermore, TSCH comes with a radio channel hopping mechanism, where the use of multiple radio channels allows the nodes to "hop" from one radio channel to another. The transition is carried out by the following pre-agreed pseudo-random algorithm:

$$\texttt{frequency} = F(\texttt{ASN} + \texttt{channelOffset}) \,\%\, \texttt{nFreq} \tag{1.1}$$

where $channeloffset$ is a "virtual channel" that is translated into a physical radio channel that is going to be used for communication. $nFreq$ is the number of available physical channels (e.g., 16 when using IEEE Std 802.15.4-compliant radios at $2.4GHz$ with all channels in use). F is a look-up table function that translates the result from the operation to an actual radio channel (i.e., from $11th$ to $26th$ in $2.4GHz$ band). In Fig. 1.4, a typical TSCH schedule is depicted.

To define a TSCH schedule, for each radio link a collection of timeslots and channel offsets is assigned, called "cells".

Finally, in a TSCH network, the nodes continuously re-synchronise on a periodic slotframe with their neighbours, based on Enhanced Beacon (EB) frames. Moreover, the EBs contain time and channel frequency information, as well as information about the initial link and slotframe for new nodes to join the network. Thus, new nodes may join a TSCH network by "hearing" an EB frame from another node.

IEEE Std 802.15.4-2015 TSCH is the baseline MAC protocol for the contributions presented in Chapter 3, and in Chapter 4.

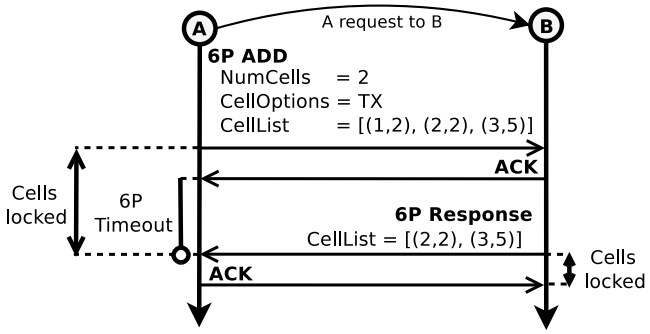


Fig. 1.5: An example of a 2-step 6top ADD transaction. The request is made from node A to node B. Node A requests two cells among three proposed candidates. Node B responds by providing two selected cells among the candidates. *Taken from* [34].

6TiSCH Operation Sublayer (6top)

IETF 6TiSCH WG [28] envisions an IPv6-based low-power wireless mesh architecture based on TSCH mode of the IEEE Std 802.15.4-2015 standard [30]. Towards this aim, it defined a new sublayer, called 6TiSCH Operation Sublayer (6top), to fill the gap between the 6LoWPAN and the IEEE Std 802.15.4-2015 TSCH layers [31]. 6top is composed of the 6top Protocol (6P) [31] and one or more SFs [32], [33].

6top Protocol (6P)

In [31], the 6P protocol is defined to support distributed scheduling in 6TiSCH networks by enabling the negotiation of cells between neighbouring nodes. More specifically, it defines the messages and transaction mechanisms to "add", "delete", or "relocate" cells within the slotframe, meaning 6P allows neighbour nodes to add/delete TSCH cells to/on one another. Additionally, it also provides commands to "count", "list", or "clear" all the cells reserved for communication between two nodes as well as a signaling mechanism for proper operation of the scheduling functions. Note that the decision of when and how many cells to add or delete is left to a 6TiSCH SF. Each 6P transaction consists of either 2 or 3 steps. In a 2-step 6P transaction, the source node selects the candidate cells, an example of a 2-step ADD transaction is illustrated in Fig. 1.5. In a 3-step 6P transaction instead, it is the destination node which selects the candidate cells.

Scheduling Function

The 6P protocol only provides the necessary transactions. A SF decides when to add/delete cells, and it triggers 6P Transactions (commands) according to application needs, routing changes or schedule collisions.

A plethora of SFs has been proposed in the literature to address different traffic patterns or to optimise a network feature [35]. Most of these SFs can be classified in the following categories:

- Centralised: A central entity in the network (e.g., Path Computation Ele-

ment (PCE)) builds and distributes a schedule by considering the gathered information by network's nodes such as network and routing topology, requested traffic or link quality.

- Distributed: Each node constructs its schedule based on the information exchanged with its neighbours.
- Autonomous: Nodes construct their local schedule, without the intervention of any central or distributed scheduling entity.

To define a TSCH schedule, for each radio link a collection of timeslots and channel offsets is assigned, called "cells".

6top is the baseline sublayer for the contributions presented in Chapter 3, and in Chapter 4.

RPL Routing Protocol

Scenarios such as smart grid or factory automation in which the product can be for instance cars, require low-power mesh networking that consist of hundreds of sensors and actuators [10], [36]. Therefore, in order to extend the network beyond the radio coverage of one node, a mesh technology enables some of the nodes to act as relays for others, but, beyond one hop, it will require a protocol for routing packets throughout the network.

RPL [25] is one of the most adopted routing protocol for the LLN. It is a proactive routing protocol specified by the ROLL WG [24] to specifically designed to manage hundreds of nodes in LLNs. RPL is defined as link-layer agnostic, so it can operate over wireless or Power Line Communication (PLC) networks for example.

The nodes running the RPL routing protocol construct a Destination Oriented Directed Acyclic Graph (DODAG) using a distance-vector technique. In each DODAG, there is a single node considered to be the border router, called the DODAG Root, which serves as the gateway to other non-RPL networks. Based on a common Objective Function (OF) [37], [38], [39] each node selects one or more parent(s), acting as a relay toward the DODAG Root. In such acyclic network, the traffic from non-root nodes towards the root is called the upstream traffic, while the reverse is called the downstream traffic.

RPL supports two modes of operation for performing downwards (root to leaves) routing:

- Storing mode: where intermediate nodes maintain state representing the routing information towards all their ancestors (direct children, children of children, etc.). This results in additional memory requirements for the nodes maintaining this information but higher network performance, especially for inter-DODAG traffic.
- Non-storing mode: where only the DODAG Root maintains state representing the routing information towards the whole DODAG. This results in high memory requirements only on the root, but adds network overhead due to the use of source routing and less efficient intra-DODAG routing because all the traffic needs to go through the root.

RPL uses three types of control packets to form, maintain and update the DODAG:

- DODAG Information Object (DIO): Contains all the necessary information regarding a RPL Instance, DODAGID, DODAGVersionNumber, or routing related metrics that will allow a node to select its parent. It is transmitted periodically in multicast by the RPL nodes. The interval of the DIO transmissions depends on the network stability. Indeed, the more stable the network, the fewer the DIO packets sent in order to reduce the overhead. Correspondingly, when the network is not stable, more DIO messages are transmitted. This timing algorithm is called a Trickle timer [40].
- Destination Advertisement Object (DAO): Propagates reverse routing information and is transmitted every time a new preferred parent is selected. It is sent as unicast and delivered to the new parent node or the DODAG Root, depending on the established RPL storing method (storing mode / non-storing mode).
- DODAG Informational Solicitation (DIS): Interrogates other nodes about the status of the network, soliciting DIO messages as a response.

Finally, to avoid loops in the DODAG, RPL performs distance-vector routing based on a rank hierarchy given by an OF. Out of a set of candidate parent nodes, the node with the lowest rank value is selected as the Preferred Parent (PP) and used for upwards routing. To calculate a rank, the OF can use different metrics [41], usually one of the two following:

- Hop Counting (HC): number of hops a packet must perform to reach its destination.
- Expected Transmission Count (ETX): Statistics aggregation that reveals a point-to-point link's quality. This metric reveals a route with the highest success probability.

RPL is the baseline routing protocol for the contributions presented in Chapter 5.

Chapter 2

Positioning

2.1 Toward Reliable & Available Low-Power Wireless Mesh Network

As stated in the previous Chapter, among the IoT-based applications, critical applications, including industrial applications, have been the core of my research during the past six years. The industrial use-case is an emerging domain of applications for the IoT [7]. It consists of re-using the IoT technologies to simplify the production chains, ease the deployment and maintenance, and make the factory more flexible and adaptable [42]. Management cost reduction and factory automation can be achieved, in particular, by replacing the existing cables with a wireless medium, as long as an appropriate level of service for critical applications can still be guaranteed at all times [43]. Indeed, network performance metrics such as throughput, Packet Delivery Ratio (PDR), and latency must be managed accurately depending on the application requirements.

To reach this ambition, the network must be "deterministic". Deterministic in this context means that the designed and deployed low-power wireless mesh network provides guaranteed bandwidth, bounded latency, and other properties germane to the transport of time-sensitive data [44], [45]. Some of the most important characteristics of the network are to exhibit bounded latency (i.e., a jitter, different on the consecutive packet inter-arrival time, close to $0ms$), or bounded number of losses-in-a-row [46]. A deterministic is a required property for instance, in the power grid, to ensure that high tension lines breakers can be activated within milliseconds, in public transportation to make sure that automated vehicles are operated safely for their passengers, and in industrial automation for control loops [42], [43]. However, the current technologies deployed for the low-power wireless mesh networks are based on best-effort packet switched networks. Indeed, the data are encapsulated within packets that are subject to variable delays in the multi-hop wireless network, due to retransmissions and enqueuing in intermediate nodes.

To further delineate the concepts behind deterministic networking, let us consider some analogies from the real-world [47]:

- Bus trajectory analogy (to deterministic circuit switching): the objective of the deployed reserved bus lanes in cities is to avoid that the bus is delayed by traffic jams and guarantee an on-time and repeatable experience for the passengers.

- Metro or train network analogy (to control loop traffic): the trains share the same infrastructure, while serving different stations and drive at different speeds, yet avoiding collisions. In addition, trains (usually) arrive on time.
- The vacation place analogy (to time-sharing): an intelligent marketing concept of time sharing acquisition whereby an individual rents a flat for just one week (e.g., for vacation or business trip purposes) a year.

While Deterministic Networking (DetNet) specifications [48] can be applied to both wireless and wired mediums, there has been recent industrial use cases for wireless applications which were not initially included in the DetNet use cases, e.g., Aeronautical Data Communications [49]. Indeed, in aeronautical data communications, there is significant interest in Internet Protocol (IP) connectivity applications. Industrial automation, pro audio and video, gaming, and edge robotics are among other examples of potential wireless applications that require deterministic solutions [43].

However, uncontrolled interference, multi-path fading and random obstacles may impede the transmissions over-the-air, causing thus rapid variations of the link quality and throughput [50]. This nature of wireless communication limits the volume and/or duration of data traffic that can be successfully transmitted on a wireless medium [51], [52], and, thus introduces considerable challenges to achieve deterministic properties such as low packet error rate, bounded consecutive losses, and bounded end-to-end latency. *Therefore, the term deterministic is usually not associated with a short range wireless communication, and, in particular one operated in the Industrial, Scientific and Medical (ISM) band.* As a result, at the IETF standardisation body, for networks that include a collection of wireless segments, they defined the term "Reliable and Available Wireless" instead of "deterministic". Indeed, a new WG called Reliable and Available Wireless (RAW) was formed to take up the challenge of providing highly available and reliable end-to-end performances in a network with scheduled wireless segments. To get around that, RAW leverages scheduling (e.g., TSCH) and all possible forms of diversity (e.g., frequency, time, space) to defeat the possible causes of transmission failure while preserving energy and optimising the use of the shared spectrum.

Therefore, increasing the reliability and predictability of low-power wireless mesh networks has been the main research challenge I have been addressing since 2016, after my PhD defense.

2.2 My contributions

In order to eventually achieve reliable, and available and bounded latency in low-power wireless mesh networking, during the last six-years of intensive research, I have focused on MAC, Scheduling and Routing layers. Since, each of them provides key-actions that enable reliable and predictable networking.

These three layers can provide solutions on the following fundamental networking questions, *when*, *how* and *whom*:

- MAC protocol: *when* and over *which* radio channel a transmitting node should "wake-up" to transmit a frame, and when and on which radio channel a receiving node should "wake-up" to receive a frame.

- Scheduling Function: *how* to *fairly* allocate the available resources, i.e., the transmitting and receiving *cells*, among the nodes in a low-power wireless mesh network.

- Routing protocol: to *whom* a transmitting node should send its IPv6 data packet, i.e., who is its next hop, in order this data packet successfully arrives to the final destination.

In the following sub-sections, I present the main research avenues that I took as well as the challenges that I had to tackle in order to achieve the previously presented objectives.

MAC Layer

I have worked for long time on performing exhaustive radio link characterisation as well as thorough evaluations of the connectivity under real-world conditions of the IEEE Std 802.15.4-2015 standard by employing various low-power devices (e.g., MSP430, Cortex M3) and network topologies [50], [51], [52]. Indeed, I have investigated the link quality, symmetry and stability under real-world environments, i.e., FIT IoT-LAB [1]. Moreover, I have evaluated the reproducibility performance over time by repeating the experiments seven times over different days and time periods of the day.

Some of the "lessons learnt" from these experimental works are: *i*) the links between the nodes are often unidirectional, *ii*) only few radio channels (e.g., 15, 20, 25) remain somehow preserved from external (i.e., IEEE 802.11) interferences. *iii*) even "good" radio links vary over time, which reflects that naive solutions for reproducibility (i.e., keeping only good links) may fail.

These results demonstrate that since IEEE Std 802.15.4-2015 standard comes with 16 radio channels, and while its TSCH mode allows for channel hopping over subsequent frame transmissions, it would be worth examining its performance when excluding the "poorly" performed radio channels from the channel hopping pre-agreed pseudo-random algorithm [19].

At first, one may consider that this approach is a straightforward and constitutes a trivial solution. However, while investigating and applying it into low-power wireless mesh networks, it might be proven rather a tricky approach. Below, I list some of the challenges that should be considered:

- **Link Quality Indicators**: based on which link quality indicators one should evaluate the radio links?

- **Threshold**: what would be threshold value, inferior which a radio channel will be blacklisted? Moreover, this threshold should be *a priori* defined or rather adaptive?

- **Global or local blacklisting**: a single radio channels should be blacklisted globally for the whole wireless mesh network, or locally per-hop?

- **Centralized or distributed approach**: a radio blacklisting decision should be taken in a centralised i.e., PCE, or in a distributed manner, i.e., a pair of nodes monitors its link quality and decides which radio channels to blacklist?

[1] https://www.iot-lab.info/

- **Consistent blacklists**: how the transmitter and the receiver will agree on consistent blacklists?
- **From blacklisting to whitelisting**: how to recover a radio channel from a blacklist to whitelist?
- **Channel hopping sequences** : how to avoid internal collisions due to potential different local blacklists among the nodes?

As I further detail in Chapter 3, in which several distributed and centralised original contributions are described, efficient radio blacklisting allows for very-high end-to-end network reliability, and, moreover it reduces essentially the number of retransmissions, which corresponds to decrease of latency.

This, radio channel blacklisting, was my ***first*** main research axis toward reliable and bounded latency. In this research field, I was privileged to co-supervise one PhD student, two Master students, and to collaborate with great academics and researchers. Together, we have published 2 journals articles (including 1 magazine article), and 6 conferences papers.

Scheduling Function Layer

Providing high *reliability* by combining IEEE Std 802.15.4-2015 TSCH with a blacklisting technique into a low-power wireless mesh networks does not mean that *bounded latency* is provided at the same time. In fact, an appropriate organization of the transmissions is required to minimize the end-to-end delay. In other words, an efficient scheduling function algorithm is necessary. The IEEE Std 802.15.4-TSCH standard does not specify how to build a TSCH radio link schedule, leaving the construction of the schedule to network administrators.

Resource allocation and SF is not a recent research field. Indeed, since 2012, the early days of 6TiSCH WG formation, a number of SFs has been proposed in the literature for low-power wireless mesh networks [35]. They are classified in the centralized [53], [54], distributed [55], [56], [57] and in autonomous [58], [59] categories. IEEE Std 802.15.4-2015 TSCH supports all centralized, distributed and autonomous scheduling functions.

Indeed, after several draft proposals at the IETF [60], [61], the 6TiSCH WG converged with 6TiSCH Minimal Scheduling Function (MSF), a distributed function, as the main SF. However, our thorough works on performance evaluation of the MSF [34], [62] demonstrate that the defined scheduling function is not mature enough, and, in particular we have observed the following issues:

- The duration to allocate the necessary resources has a direct impact on the amount of losses seen during traffic load changes. Moreover, the rate at which the resources are allocated depends on the number of cells already allocated in the slotframe.
- MSF is subject to over-provisioning of the network resources, and frequently allocates or keeps more cells than are required to send the current traffic load.
- The degree of customisability offered by MSF is limited, which can lead the protocol to unstable behaviour.

Therefore, after eight years, a design of an efficient SF still remains very challenging and popular research field. Furthermore, considering that MSF comes only with best-effort features, and it is still far from achieving the objectives of ultra-reliable and bounded latency networking. Therefore, in early 2020, a new WG called RAW [63], was formed with a focus on rather centralised approach, where I have been contributing with several Internet Drafts (I-Ds) [45], [64], [65].

As I further elaborate in Chapter 4, I had also the privilege to also contribute to on non-standardisation track works. Indeed, I was involved in both distributed and centralized original research proposals. Notably, our distributed proposal, Low-latency Distributed Scheduling Function (LDSF) [29], achieves low latency and jitter with high network reliability, even when the radio links are unreliable, and it outperforms most of the state-of-the-art proposals such as MSF [33], Low Latency Scheduling Function (LLSF) [56] and Stratum [57].

Our centralised approaches, based on Linear Programming (LP) and Simple Descent (SD) [21], achieve as well great performances, and most importantly even when applying them over wireless nodes that are placed inside an Electric Vehicle (EV) battery pack to enable wireless Battery Management System (BMS).

Designing and developing SFs was my ***second*** main research axis toward reliable and bounded latency. In this research field, I was privileged to co-supervise two PhD students, and to collaborate with great academics and researchers. We have published 3 journals articles, 2 conferences papers, 1 poster, and 3 I-Ds submitted (where the 2 are adopted) for standardisation at the IETF.

Routing Layer

To recall, "determinism" (reliability and predictability) in a network means the guarantee that a particular data packet is transported across the network in a bounded window of time. It also guarantees that a periodic process will be repeated identically throughout the network lifetime, where potential congestions or external interferences must not affect the reliable and predictable properties of the network.

However, combining IEEE Std 802.15.4-TSCH with a blacklisting technique in conjunction with robust SF into a low-power wireless mesh networks does not mean that ultra-high end-to-end network *reliability* and *bounded latency* is guaranteed under any network conditions. Indeed, wireless links are subject to losses, due to low link quality (i.e., multi-path fading, distance), external interferences, or outage of one node which may decrease the reliability [51]. Thus, wireless transmissions are typically associated with a retry mechanism, such as Automatic Repeat reQuest (ARQ) [66]. Retransmission comes at a cost in terms of delay, energy consumption and bandwidth, since it requires additional timeslots under TSCH MAC. Typically, if a transmission of a frame fails, the transmitting node may need to wait for several timeslots, or even for the next slotframe to retry, which reflect rather a best-effort traffic. In some cases, frame losses can even be bursty, for instance when the link quality between two nodes is drastically reduced for a certain time window, or when there is outage of one of the two nodes. In such scenario, retransmissions will not allow for a frame to pass through this wireless link. The standardised RPL routing protocol proposes failover mechanisms (i.e., Local and Global repair), but the delay to discover and use an alternative path is too high [67].

Considering such challenges, as I further describe in Chapter 5, I had the privilege to contribute both with innovative proposals in academia and with I-D at the IETF standardisation body [39]. The conducted work was employing RPL as the baseline, and extends it with multi-path redundancy, which exploits data packet replication & elimination, promiscuous overhearing, and Forward Error Correction (FEC) [68] to combat isolated and cumulated losses in the network. The thorough performance evaluation campaign show that reliability and predictability in low-power wireless mesh networks can be guaranteed by using multiple parallel data paths instead of retransmissions along a default single path [13], [15], [14], [16], [17], [69], [18], [70].

These multi-path routing algorithms, was my ***third*** and last main research axis toward reliable and bounded latency. In this research field, I was privileged to co-supervise one Postdoctoral fellow, one PhD student, four research interns, and to collaborate with great academics and researchers. We have published 2 journals articles, 6 conferences papers, 1 poster, 2 national conferences papers, and 1 I-D which is submitted for standardization at the IETF.

2.3 Manuscript Organization

I organize the core of this manuscript in three chapters, Chapters 3 to 5, each covering a specific layer from the protocol stack which I believe are the key enablers to achieve high end-to-end reliability and bounded latency. I have focused on MAC, scheduling function and routing layers. The chapters are organized so they present a logical progression, which reflects the six-years of my journey through this challenging research field that I am so passionate about.

Chapter 3 focuses on the MAC layer, and, in particular on the IEEE Std 802.15.4-2015 TSCH protocol. It tackles the research problem of radio channel blacklisting and whitelisting of the total 16 available in $2.4GHz$ band. Several distributed and centralised proposals (along with global and local ones) are presented, while a thorough discussion on their trade-offs is followed. The order of the sections (in fact the proposed algorithms) reflect a chronological progression, which in turn reflects the maturity of the proposals.

Chapter 4 tackles with the SF algorithms. Both distributed and centralised research works in which I have contributed are presented. Performance evaluation results followed by a trade-off discussion on distributed versus centralised approaches is given. I continue working till today in this research field with two PhD students, who started in 2020. More detailed inputs will be given in the perspectives section of chapter 6.

Chapter 5 covers the contributions that have been conducted at the routing layer that extend the RPL protocol to support multi-path routing. Indeed, several redundant routing algorithms and protocols are presented. Once again, performance evaluation results along with trade-off discussion on single-path versus multi-path, disjoint versus common ancestor algorithms, and network reliability and bounded latency versus energy consumption and network capacity is followed. The order of the sections reflect a chronological progression, which in turn reflects the evolution of the proposed redundant multi-path algorithms.

Finally, Chapter 6 provides concluding remarks and a summary of the main lessons I have learnt so far, and presents the envisioned perspectives of high-risk-high-gain research which I believe are worth to pursue.

Chapter 3

Radio Channel Blacklisting Techniques in TSCH-based Wireless Mesh Networks

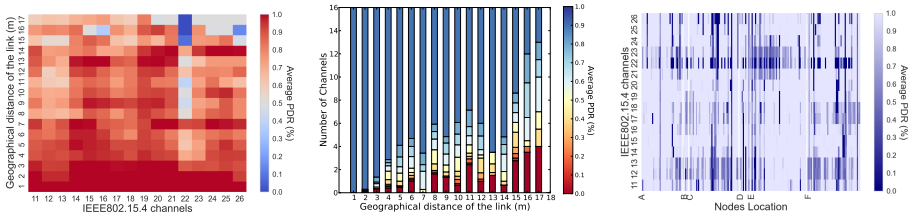
The continues growth of the low-power wireless constrained networks, which exploits wireless technologies that employ the same frequency band, i.e., $2.4GHz$ ISM band, results in a large concentration of wirelessly operated devices in the same area, causing thus intra- and inter-technology interference [51], [52]. Another cause that heavily contributes to the unreliability of wireless radio links is the multi-path fading effect. The multi-path fading effect can be caused when the radio signal arrives the receiver through multiple paths due to reflection off obstacles [71]. In industrial wireless networks, the multi-path fading effect is magnified due to highly reflective structures such as metallic objects in the industrial environment.

In this Chapter, we will first tackle the question *whether or not radio channel blacklisting technique when applied over slow channel hopping protocols is a relevant approach toward mitigating the external interference, and the multi-path fading effect.* Moreover, we will present series of original blacklisting (and whitelist) proposals that we have been put forth during the ***PhD period of Vasileios Kotsiou*** [19].

To this end, a thorough experimental characterisation of the 16 IEEE Std 802.15.4-2015 radio channels and connectivity among the nodes of an indoor testbed was conducted. We investigated the spatial and temporal characteristics of the wireless links, and the diversity in radio link's quality among the radio channels. The experiments were conducted over the FIT IoT-LAB testbed [1], which belongs to the half real-world testbed category since several Wi-Fi Access Points are co-located [50]. Moreover, OpenWSN was employed as the operating system, which comes with the implementation of the 6TiSCH stack [8].

Our experimental results highlight the existence of specific per-link characteristics, where external interference may be locally high for some radio channels, see Fig. 3.1a. More specifically, the results demonstrate the fact that the quality of the radio links is distance-dependent, see Fig. 3.1b, location-dependent,

[1] https://www.iot-lab.info/



(a) Heatmap: average link quality through each channel depending on the radio link geographical length.

(b) Classification of the link qualities for different geographical lengths.

(c) Heatmap of the PDR for each channel, links being grouped by their geographical location.

Fig. 3.1: Variability of the list of *bad* channels. *Taken from* [72].

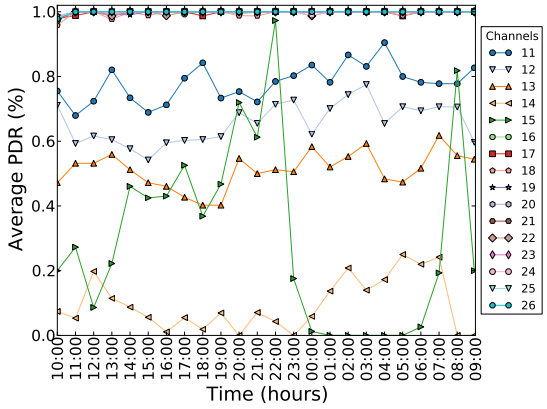


Fig. 3.2: The variability of the link quality over time: case of 9 m distance. *Taken from* [72].

see Fig. 3.1c, and varies over time, see Fig. 3.2. Therefore, in order to improve the efficiency of the slow channel hopping technique, applying a radio blacklisting technique where the "poor" radio channels are excluded from the channel hopping sequence seems promising [72].

Towards this aim, *i*) we first proposed a distributed link-based blacklisting technique, that adopts a pseudo-random approach to prevent using the low-quality radio channels [73]. However, while this approach is based on collision-free MAC protocol such as the TSCH, internal collisions may still present, since each pair of nodes (each radio link) select autonomously the "best" radio channels to use. *ii*) Therefore, we then worked on a centralised approach, and proposed a solution that is able to adapt the whitelists for each radio link by re-arranging the conflicting whitelists, and, thus achieving a rather more reliable and predictable networking [74], [75]. *iii*) Finally, we have also proposed an hybrid blacklisting technique that exploits the full radio spectrum by assigning all the channel offsets to increase the network efficiency when handling long blacklists [76].

3.1 Distributed Blacklisting Technique

In the previous section, we investigated the characteristics of a low power lossy network in an indoor environment. In particular, we highlighted the existence of low-quality radio channels. Therefore, in this section, we propose to investigate how we can use a blacklisting technique to improve the network reliability. Blacklisting techniques identify the *bad* radio channels, and exclude them from the channel hopping sequence in order to not using them to transmit data packets. In this way, we reduce the number of unnecessary transmissions, with a positive impact on both the reliability and the duty-cycle ratio.

Towards this aim, in this section, we present our first contribution called Link-based Adaptive BLacklisting (LABeL), a distributed per radio link blacklisting technique [73]. The proposed distributed blacklisting technique is flexible to network dynamics, and is able to respond directly to variations in radio channel quality by adapting the blacklist accordingly, while the overhead to the network traffic is limited. Moreover, the transmitter independently assesses the available radio channels and dynamically selects the best of them to be used in the channel hopping sequence. Finally, the transmitter and the receiver agree on a consistent blacklist to avoid *deafness*, meaning when the transmitter is sending a data packet, the receiver will actually "listen" on the agreed timeslot and radio channel to receive the transmitted data packet.

Deciding which Radio Channels to Blacklist

According to our work in [72], relying on Received Signal Strength Indicator (RSSI) or Link Quality Indicator (LQI) metric is not representative of the channel quality. Therefore, we focused rather on measuring the PDR performance, denoting accurately the ability of the link to deliver successfully the data packets.

To this aim, each node in a TSCH network computes the PDR of unicast data packets **independently** for each neighbour and radio channel. More precisely, a node counts the number of acknowledgments (ACKs) and the number of packets transmitted to a particular neighbour.

Many proposals consider a fixed threshold value [77], [78], where any radio channel that provides a PDR inferior to a pre-defined threshold value is blacklisted. However, in this study, we focused on an **adaptive** approach in which the threshold depends on the runtime link quality, and is not fixed a priori globally.

The Window Mean Exponentially-Weighted Moving Average (WMEWMA) has been proved to accurately estimate the link quality [79]. Indeed, packet losses represent a stochastic variable and need to be *smoothen*. We consequently propose to use WMEWMA to independently measure the PDR for each radio channel. In this proposed work, each node computes the PDR for the last 16 transmitted packets for a given radio channel, and updates accordingly the smoothed PDR metric.

Constructing a link-based blacklist requires only for the transmitter to collect the ratio of acknowledged packets. In particular, the blacklist considers both directions, for respectively the data packet and the acknowledgement transmissions. Thus, computing the blacklist does not need to send explicit control and probe packets, and does not generate any overhead.

Modifying the Frequency Hopping Sequence

After identifying the blacklisted radio channels, we next have to exploit this blacklisting mechanism with TSCH. Indeed, the employed physical channel is determined at the beginning of each cell of the schedule, using Eq. 3.1.

$$frequency = F\Big(\Big(ASN + channelOffset\Big)\ \%\ nFreq\Big) \tag{3.1}$$

However, since some of the 16 radio channels might be blacklisted, the execution of the radio channel hopping algorithm may result in a blacklisted radio channel. Therefore, we proposed to adapt the frequency hopping method in order eventually to provide a whitelisted radio channel. More specifically, our proposed radio channel hopping algorithm makes the distinction between the following cases:

C1: **Good radio channel**: If the physical channel is not blacklisted (i.e., whitelisted), let's use it;

C2: **Blacklisted radio channel**: If the physical channel is blacklisted, let's select pseudo-randomly a good radio channel. The function must use a common knowledge between the receiver and the transmitter to avoid deafness. We proposed to select the radio channel accordingly:

$$frequency = F\Big(\Big(ASN + channelOffset + k\Big)\ \%\ nFreq\Big) \tag{3.2}$$

with k the minimum integer value such that "frequency" corresponds to a *good* channel. Since ASN, $channelOffset$, $nFreq$ and the blacklist are common to the receiver and the transmitter, they will lead to a consistent decision.

Modifying the Channel Hopping Sequence to Passively Monitor the Quality of Bad Channels

We continuously estimate the PDR performance for all radio channels, including the blacklisted ones. Indeed, since the radio conditions may change during the deployment [50], [80] [51], [72], we should recover a radio channel from a blacklist to whitelist, when its PDR performance exceeds the given threshold value. However, dedicating resource (control packets) to probe bad radio channels is not recommended since it would be costly in terms of energy consumption and traffic. Note that in such a case, the probe has to be done for each blacklisted radio channel for each radio link.

In this study, we therefore proposed to monitor the link quality using a passive method, exploiting directly the reliability statistics of data packets. However, a bad radio channel should be probed less frequently than a good radio channel since it has a negative impact on both the reliability and the energy consumption.

Thus, we modified the previous second rule (C2) when computing the channel hopping sequence. More precisely, when Equation 3.1 returns a blacklisted channel:

C2.1: With the probability p, let's transmit the packet through this *bad* channel to keep on re-estimating the link quality for *all* channels;

C2.2: Otherwise, the transmitter and receiver select pseudo-randomly a good channel, applying the original C2 rule.

A small p value means that the blacklisted channels will be probed infrequently. Re-estimating the quality consumes less resource, but requires a longer time to detect radio link quality change.

Toward Consistent Blacklist between the Transmitter and the Receiver

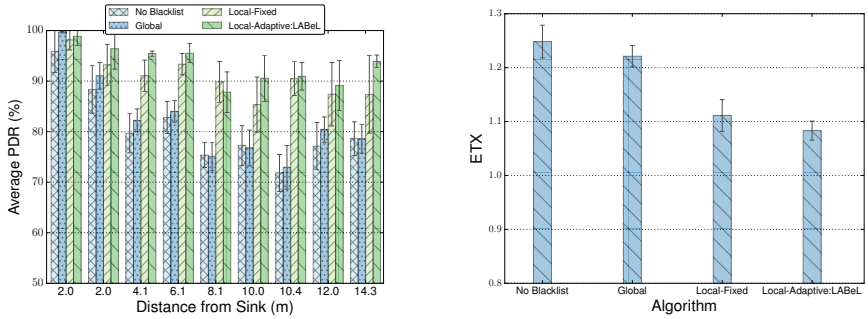
Finally, considering the nature of the distributed system, we should ensure that the transmitter and the receiver have the same blacklists, else they would use a different pseudo-random channel hopping sequence, leading to a deafness. Thus, to this aim, the transmitter sends to the receiver its blacklist using a reliable method since the receiver is not aware of the actual statistics computed by the transmitter, and cannot construct the same blacklist. We proposed to exploit 6P to exchange the blacklists for each radio link ($e.g., AB$). More precisely, the transmitter A sends its blacklist in a 6P control packet. Since the 6P packets are transmitted through the shared cells and are prone to collisions, and therefore B needs to send an acknowledgement. As a result, the use of consistent blacklists for both sides is guaranteed. Therefore, there should be a mechanism to guarantee consistent blacklists for both sides.

Experimental Performance Evaluation

We conducted our experimental campaign over the FIT IoT-Lab testbed, where the nodes are subjected to external interference originated from Wi-Fi-based devices. We focused on a 1-hop star topology scenario with 10 M3 nodes, based on a STMicroelectronics 32-bit ARM Cortex-M3 micro-controller, with various distances between the transmitters and the receiver in order to focus on the performance of a given radio link. We performed 120 experiments, while each experiment lasted for 120 min. The transmitter (leaf) node implements a Constant Bit Rate (CBR) application model, by transmitting 1 data packet every 3 $seconds$, at 0 dBm transmission power, resulting in more than 2000 $pkts$ transmissions in total per experiment. To conduct our experiments, we employed a modified implementation [81] of OpenWSN[2]. The details of the setup are presented in [73].

We compared LABeL against *i)* **Default** approach where the TSCH network operates in its standard mode, and uses only the channel hopping technique to defeat external interference, *ii)* **Global Blacklisting** where the three radio channels that are the most impacted by the interfering Wi-Fi networks are globally, for the whole TSCH network, blacklisted, and *iii)* **Local-Fixed** where all radio channels that exhibit a PDR lower than a fixed threshold value are blacklisted.

[2]branch *"track"* of `https://github.com/ftheoleyre/openwsn-fw/` and `https://github.com/ftheoleyre/openwsn-sw/`



(a) Packet Delivery Ratio.

(b) Average number of transmissions before receiving an ACK.

Fig. 3.3: Per link PDR (left) and ETX (right) performances. *Taken from* [73].

We measured the PDR performance for each radio link, see Fig. 3.3a. In overall, all blacklisting techniques improve to some extent the PDR performance. For short distance (and strong) links, PDR is very high (above 95%) independently the employed blacklisting technique. However, the blacklisting techniques improve slightly the PDR, even for these strong links. On the other hand, the weaker radio links tend to be more sensitive to external interference [72], and, thus the poor radio channels, under high level of external interference, are negatively effected in terms of network reliability. The global blacklisting provides the lowest improvement, while the *Local blacklisting with a fixed threshold* value is also suboptimal, since weak radio links tend to exhibit a low average PDR for the majority of its radio channels, see Fig. 3.1b. Thus, a medium PDR does not mean that a radio channel should be blacklisted. Finally, LABeL that computes dynamically the threshold value for the PDR, according to the best channels, is more effective to blacklist only the less efficient radio channels.

Next, we measured the ETX which reflects to the delay performance and the energy efficiency, see Fig. 3.3b, since a node has less packets to transmit on average to deliver correctly a data packet. As can be observed, LABeL, the link-based adaptive scheme, provides an ETX below 1.1, making on average links more robust (14% less transmissions compared to without blacklisting).

3.2 Centralized Whitelisting Technique

As presented in the previous section, the channel hopping in conjunction with blacklisting technique improves significantly the reliability of a wireless network. However, the use of local (per-link) blacklisting technique on multi-hop wireless sensor networks may introduce *internal collisions*. More precisely, *internal collisions* may caused by the interfering links that are scheduled at the same timeslot, and since they use different blacklists, collisions may arise even if different channel offsets have been assigned to them.

Therefore, we then proposed a centralised approach to use heterogeneous blacklists while maintaining a fully internal collision-free schedule. We will use in this section the term **whitelisting** instead of the blacklisting, to highlight the fact we use a "safe" radio channel allocation process, i.e., collision-free.

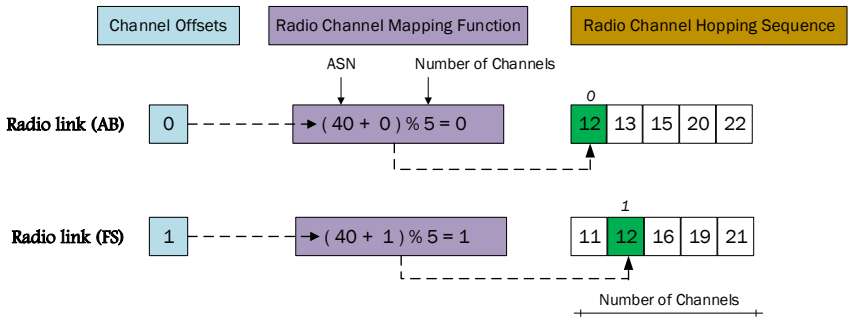


Fig. 3.4: Internal collision example when employing different whitelists. *Taken from* [74].

Problem Statement: Internal Collisions Issue

A priori, the scheduling algorithm does not introduce collisions when allocating different timeslots to the different links. Moreover, allocating two different channel offsets for two transmitters during the same timeslot is also safe *if we do not use whitelists*. The radio channel mapping function, Equation 3.1, will derive different values and, thus, different radio channels for the two transmitters to employ. However, when two or more transmitters use the same timeslot and whitelisting is considered, then an internal collision may take place. More specifically, when the whitelists comes with different lengths, or different radio channel order, and they have at least one common radio channel, then an internal collision may take place.

Let us consider two radio links (AB) and (FS) within their propagation range. The communication of these links have been scheduled in the same timeslot, but over a different channel offsets (0 and 1, respectively). Fig. 3.4 illustrates the radio channel mapping function for a specific timeslot (ASN=40), using Equation 3.1. Let us assume that (AB) and (FS) have the radio channel hopping sequence,i.e., whitelists, {12, 13, 15, 20, 22} and {11, 12, 16, 19, 21}, respectively. Upon the execution of the pseudo-random algorithm, the link (AB) has to use its first whitelisted radio channel (i.e., 12), while the link (FS) has to use the second one which is also the radio channel 12. Thus, a collision will take place in the cell at the ASN 40. We define such collisions as *internal* since they are caused by simultaneous transmissions of nodes from the same wireless network.

In [74], we have proved that two interfering radio links (l_1, l_2) that are scheduled at the same timeslot, with different channel offsets and they have one common radio channel in their whitelists may collide every:

$$\frac{LCM(|l_1|, |l_2|)}{GCD(|l_1|, -|l_2|)} slotframes \tag{3.3}$$

where $|l_1|$ is the whitelist size of the radio link l_1, $|l_2|$ is the whitelist size of the radio link l_2, LCM stands for Least Common Multiplier, and GCD for Greatest Common Divisor.

The use of a *distributed blacklisting technique* to eliminate internal collisions requires that the nodes are aware of the interfering radio links scheduled in the same timeslot as well as the employed channel hopping sequence. However, the exchange of information between 1-hop neighbours is not sufficient to acquire the above information, while message exchanges between nodes with distance greater than 1-hop increases essentially the network traffic load. Indeed, it seems more efficient to propose centralised approaches where a central entity is aware of the whitelist of each radio link, the central schedule, and the set of the interfering radio links. In the following, we present a *centralised scheduling algorithm* that assigns the cells (timeslot and channel offset) without generating internal collisions. Towards that aim, the whitelists are re-arranged when internal collisions are detected to avoid any inconsistent configurations in the radio channel hopping sequences.

A Centralised Whitelist ReOrdering Technique to Guarantee Internal Collision-free Operation in TSCH-based Network

We then opted for a centralised approach to address the previously presented issue of internal collisions by reordering the whitelists. Let us consider again the example depicted in Fig. 3.4. If the whitelists of (AB) and (FS) are reorder into {12, 13, 15, 20, 22} and {12, 11, 16, 19, 21} respectively, a collision cannot anymore happen. Indeed, a collision occurs only if the Equation 3.1 results in the same integer value. However, this will never occur since the two links have different channel offsets.

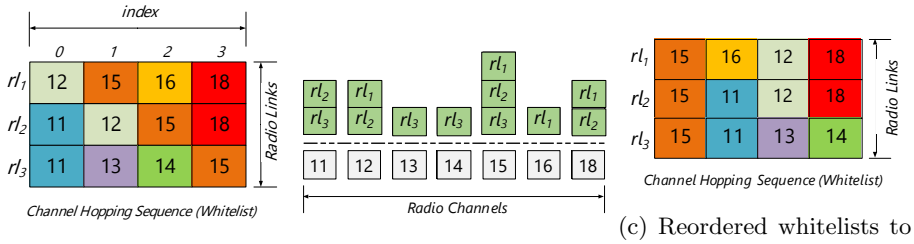
In such centralised system, the controller that is in charge for the schedule has the full knowledge of the radio topology and the radio link qualities. Therefore, when allocating two or more radio links in the same timeslot, it checks that their corresponding whitelists do not lead to collisions.

This *reordering problem* is closely related to the *University Course Timetabling Problem (UCTP)* [82], where a set of lectures have to be scheduled for a set of students, during the same timeslots. Similarly, the links correspond to the students, the channel offsets to the timeslots, while the radio channels to the lectures. Furthermore, a lecture should be given in a single timeslot, for all students who have to attend it. Similarly, a radio channel must be located at the same place in the different channel hopping sequences.

The above problem is NP-complete, and the research community has proposed many algorithms to address it such as genetic, hybrid, tabu approaches [83]. In our case, the channel hopping sequence length and the number of links during the same timeslot are reasonably small. Thus, a greedy approach seems acceptable to produce an efficient common whitelist.

At a glance, the input of our algorithm is a matrix where each row corresponds to the whitelist of a link scheduled in a given timeslot (Fig. 3.5). The output of the algorithm is the same matrix, with the reordered whitelists to avoid collisions.

For instance, let us assume three radio links have their channel hopping sequence lists, i.e., whitelists, as depicted in sequential order in Fig. 3.5a. These whitelists have to be re-arranged, else, the radio channel 15 for instance, which is used by all three links, may introduce collisions. Thus, we proposed, first to



(a) Whitelists of the three radio links in the same timeslot

(b) Stacks of the links which have each radio channel in their whitelist

(c) Reordered whitelists to exploit consistent (i.e., non colliding) frequency hopping sequences

Fig. 3.5: Reordering process of the whitelists for a group of links scheduled during the same timeslot. *Taken from* [74].

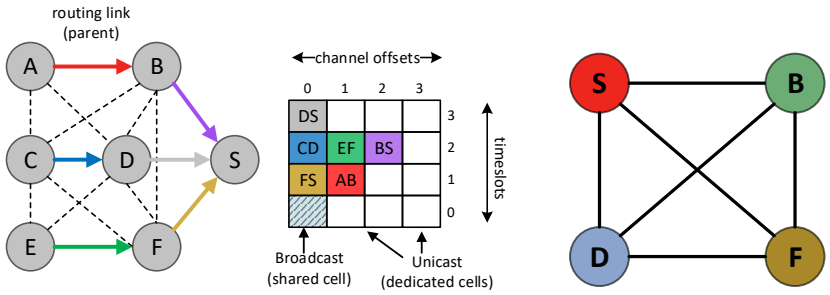
construct the radio list of links associated with each radio channel (Fig. 3.5b), then, to apply our greedy algorithm, see section IV.B in [74], which selects the radio channels in the following order: *i)* it first identifies the most "popular" radio channel(s), which is 15 in our example, and placed it at the beginning of all whitelists. *ii)* then, the less frequent radio channels are selected, i.e., 11, 12 and 18 in our example, that are used by two radio links. These radio channels are placed in the reordered channel hopping sequence sequentially, first the radio channel 11, then 12, and finally 18. *iii)* as a last action, since the first link does not have the radio channel 11 in its whitelist, it has to identify a radio channel that is whitelisted only by this radio link, e.g., the radio channel 16. As a result, the proposed greedy algorithm provides a reordered collision-free solution, as illustrated in Fig. 3.5c.

3.3 Hybrid Blacklisting Technique

In the previous section, a centralised technique was presented which on the one hand, eliminates internal collisions but, on the other hand, do not adapt the radio links' whitelists to the external interference's time variations.

Therefore, we worked on an hybrid approach that extends the state-of-the-art Multi-hop And Blacklist-based Optimised TSCH (MABO-TSCH) [84] blacklisting technique. MABO-TSCH is a combination of a centralised algorithm, where some decisions are taken distributively. Indeed, it computes the to-be-allocated cells for each radio link in a centralised manner, and assigns a collection of channel offsets to each receiver node. Thus, the transmitter can decide distributively which radio channels should not be used for the data transmissions. However, while MABO-TSCH guarantees internal collision-free operation, it is inefficient under dense wireless networks, long blacklists, and under heavy traffic load.

Therefore, we proposed an enhanced version, called Adaptive MABO (AMABO), where the allocation of the channel offsets takes place dynamically at each timeslot according to the number of interfering parallel transmissions, while still avoiding internal collisions [76].



(a) IEEE Std 802.15.4-2015 schedule for a 7 nodes topology.

(b) MABO-TSCH interference graph per ***receiver***.

Fig. 3.6: 7 nodes IEEE Std 802.15.4-2015 schedule (left) and the resulting MABO-TSCH interference graph (right) of this topology. *Taken from* [76].

Problem Statement: Receiver-based Multiple Channel Offsets Assignment

MABO-TSCH uses a receiver-based channel offset assignment, where a set of channel offsets is assigned to each receiver node. MABO-TSCH applies a graph-colouring approach, where the vertices are the nodes, the edges are the interfering links, and the colours are the 16 available TSCH channel offsets. Note that two nodes (vertices) are joined with an edge if they are neighbours, or neighbours of their neighbours. Then, an extended version of the Welsh-Powell algorithm [85] is applied to assign multiple non-interfering radio channels to each node.

Let us consider the network topology and the schedule depicted in Fig. 3.6a. Since MABO-TSCH is receiver oriented, we consider only the four different receivers, i.e., S, B, D, and F. The interference graph (which pairs of receivers mutually interfere) is represented in Fig. 3.6b. In our case, all the receivers are neighbours of the DODAG Root S, and they interfere among themselves, which corresponds to a full graph. Then, MABO-TSCH assigns centrally to each receiver multiple channels offsets different from all its interfering nodes, i.e., receivers, as illustrated in Fig. 3.7. In particular, all radio links toward S use the channel offsets 0 to 3 over different timeslots, and all different receivers use non overlapping channel offset ranges.

Besides, the number of channel offsets and the blacklist size impact directly the performance of a MABO-TSCH schedule. In particular, if the blacklist size exceeds the number of channel offsets, the radio link may not be able to always use a non blacklisted radio channel. In that case, a blacklisted radio channel needs to be employed, impacting thus negatively the reliability. As a result, **assigning a fixed number of channel offset is inefficient**, and does not capture the whole network heterogeneity.

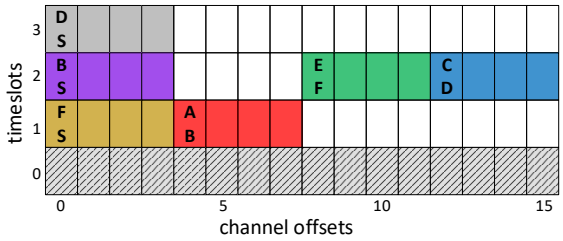


Fig. 3.7: An example of MABO-TSCH schedule, where multiple channel offset are assigned per receiver. *Taken from* [76].

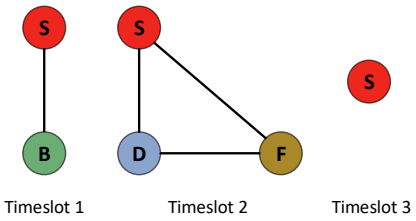


Fig. 3.8: AMABO interference graphs per ***timeslot***. *Taken from* [76].

AMABO: Timeslot-based Multiple Channel Offset Assignment

We then proposed to enhance MABO-TSCH, i.e., AMABO, by **not** assigning a fixed number of channel offsets per receiver [76]. In fact, we proposed to assign the channel offsets on a timeslot basis:

1. First, for each timeslot, we constructed the interference graph corresponding to these radio links.
2. Then, we assigned fairly the set of channel offsets to each radio link of the clique in the interference graph.

Indeed, having non-allocated cells in the schedule has no practical interest for the network, e.g., see the timeslots 4 to 15 for (DS) in Fig. 3.7. While some radio bandwidth would be available, no radio link can exploit it, even if it has a long blacklist because of a high level of external interference.

Let us consider the same topology as previously, i.e., Fig. 3.6a. According to AMABO, to each timeslot corresponds a set of duplex-conflict free[3] [86] but interfering radio links, i.e., a subgraph as illustrated in Fig. 3.8. For instance, during the first timeslot, two different links are scheduled: (AB) and (FS). To mitigate the waste of channel offsets, AMABO assigns half of the channels offsets to (AB), and the other part to the link (FS). On the contrary, the link (DS) during the third timeslot, receives all the channel offsets, see Fig. 3.9.

[3]A node n_i cannot transmit and receive at the same time, and it cannot receive from multiple nodes at the same time [86].

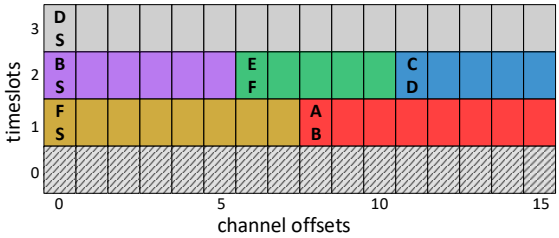


Fig. 3.9: AMABO multiple channel offset assignment. *Taken from* [76].

3.4 Performance Evaluation

Simulation Setup & Experimental Dataset

In order to evaluate all the previously presented blacklisting approaches, i.e., distributed, centralised, and hybrid, we emulated a network using an experimental dataset. Indeed, we relied on a dataset obtained from the FIT IoT-LAB testbed. More precisely, we selected a large set of radio links, which forward 1 data packet every 3 seconds. The radio links are scheduled in different timeslots to guarantee collision-free process. We stored the transmission failure/success for each data packet for a duration of 90 min[4].

We employed the previously presented dataset as an input in a custom made simulator based on Python. We then emulated a 60 node topology, plus a border router which collects the data packets that were randomly positioned in an area of 200 X 200 m^2. The radio transmission range of each device was 50 m. Then, each *emulated* link was mapped to a *real* link in the testbed. Note that we considered both the correlation among links which are geographically close, and the strength of the links (i.e., longer links tend to be statistically weaker).

We then used the success/loss event of each real link for the emulated links, while preserving the correlations for geographically close links.

Scheduling & Blacklisting/Whitelisting Algorithms

In this campaign, we employed the Traffic Aware Scheduling Algorithm (TASA) [87] to construct the schedule. At the beginning of each slotframe, each node generates a random number of data packets per slotframe in the range $[1, 5]$. We considered a slotframe size of 293 timeslots with 16 channel offsets, to be able to forward all data packets. Because we focused on the efficiency of the whitelisting mechanism, and not on the scheduling process, we did not provision additional cells for the retransmissions.

Each algorithm selected the k best radio channels to be included in the whitelist. We compared the following approaches:

Default (No Whitelisting): IEEE Std 802.15.4-2015 network operates in standard mode where all 16 available radio channels.

Global Whitelisting: each radio link ranks its radio channels according to their PDR. Then, the global whitelist selects the k best radio channels.

[4]The dataset is freely available for the research community at https://github.com/vkotsiou/grenoble-multichannel-dataset

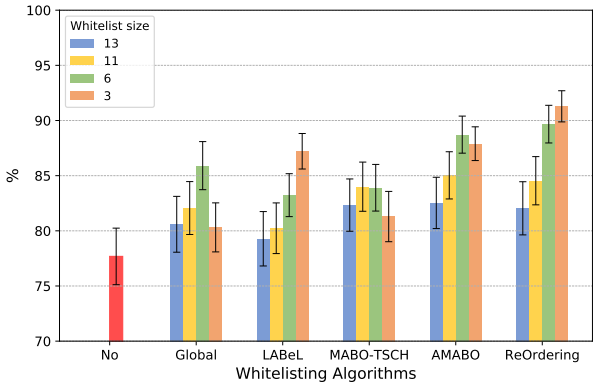


Fig. 3.10: Link-level Packet Delivery Ratio. *Taken from* [74].

LABeL: *our proposed work*, where distributed link-based whitelists are implemented, see Section 3.1.

MABO-TSCH: the hybrid approach, where the centralised controller assigns a fixed number of channel offset on a receiver basis [84].

AMABO: *our proposed hybrid work*, where the centralised controller assigns multiple channel offsets on a timeslot basis, i.e., Section 3.3.

ReOrdered Whitelists: *our proposed work*, the centralised reordering whitelist algorithm, see Section 3.2.

The detailed configuration of the evaluated algorithms is presented in [74].

Simulation Results

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [19], [72], [73], [76], [74], [75] the detailed performance evaluation results are demonstrated.

Fig. 3.10 shows the performance of all algorithms in terms of PDR. As it can be observed, TSCH without applying a blacklisting/whitelisting technique achieves the worst reliability, possibly because many data packets are transmitted over radio channels with a poor PDR performance. A *Global blacklisting* approach improves slightly the reliability by removing globally the worst radio channels. However, some of the whitelisted radio channels provide a lower PDR for *some* links. Next, the performance of our proposed LABeL algorithm, is proportional to the size of the whitelist. However, this reliability improvement comes with a decrease of the network capacity, since the traffic load has to be spread across a smaller number of radio channels.

MABO-TSCH presents better performance than LABeL in scenarios with *small size of blacklists* since LABeL introduces internal collisions. However, its performance degrades when the size of the blacklist increases since it assigns inefficiently the channel offsets per receiver and, thus, the nodes have to employ "poor" radio channels to transmit. Our proposed hybrid AMABO algorithm improves the MABO-TSCH scheme essentially, as it demonstrates reliability performance close to 90%. Indeed, it assigns dynamically per timeslot

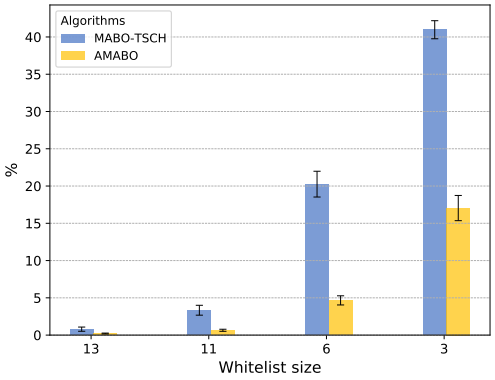


Fig. 3.11: Percentage of packets that are transmitted through a "bad" radio channel. *Taken from* [74].

the channel offsets, instead of uniformly and statically per receiver. As a result, it reduced by more than 50% the selection of "poor" radio channels when compared against MABO-TSCH, see Fig. 3.11. However, even with AMABO, when long blacklists (or short whitelists) are used, then the PDR performance drops, see the last column of AMABO in Fig. 3.10.

Finally, our proposed *centralised whitelist technique* presents the best results since radio links are using only their best radio channels at each iteration. *ReOrdering* handles efficiently the small and heterogeneous whitelists without introducing internal collisions.

Distributed schemes such as LABeL generates internal collisions, which makes the network non-deterministic, and less suitable for critical applications. On the other hand, employing ReOrdering technique is only relevant if the radio channel conditions do not change frequently. In our performance evaluation campaign, the same whitelist were used during each simulation execution, which reflected to high PDR performance. Thus, applying a centralised approach when the environment is sufficiently stable seems reasonable. Finally, when the network conditions are dynamic, contrary to a centralised algorithm, an hybrid approach such as AMABO may adapt faster, since the whitelist updates takes place distributively. However, when short whitelists are employed, there is high probability to use "bad" radio channel.

3.5 Summary

The purpose of this Chapter was to provide reliable communications to wireless mesh networks. Indeed, to support critical applications, we need to make the wireless infrastructure reliable. Considering that external interference and multi-path fading are major causes of unreliability, consequently, we focused on improving the MAC mechanisms to be more robust in this kind of environments.

To bootstrap our research, we conducted a thorough experimental study to characterise the IEEE Std 802.15.4-2015 radio channels over the FIT IoT-LAB testbed [50] by employing OpenWSN [8], an open-source implementation of the 6TiSCH stack. We studied in-depth the spatial and temporal characteristics of the radio links and the diversity in radio link's quality among the radio channels. Our objective was to investigate the relevance of using black-

listing/whitelisting techniques to improve the network reliability. The role of a blacklisting/whitelisting technique is to evaluate the available radio channels of the channel hopping sequence, to identify the low-quality radio channels ("bad"), to distribute the list of the bad radio channels to the appropriate nodes, and to exclude them from the channel hopping sequence.

We first proposed a ***distributed*** blacklisting technique, called Link-based Adaptive BLacklisting (LABeL). LABeL evaluates the quality of the radio channels of each radio link in a distributed manner and exploits an adaptive threshold algorithm to select the best radio channels for the data transmissions. Furthermore, we developed techniques based on 6P [31] control packets to ensure blacklist consistency among the transmitter and the receiver of a radio link to avoid deafness. Finally, LABeL introduces a channel hopping modification technique to re-evaluate the low-reliability radio channels without using any control message, thus saving energy. Our thorough experimental evaluation based on OpenWSN and FIT IoT-LAB highlights the relevance of this approach.

Next, we proposed a ***centralised*** whitelisting technique to eliminate internal collisions, called *Whitelist ReOrdering*. The main drawback of exploiting a local link-based blacklisting technique is the presence of internal collisions caused by the interfering links scheduled at the same timeslot while using different blacklists. These internal collisions are prejudicial to the reliability, and even worse, may exhibit a repetitive pattern. Thus, we investigated how a centralised algorithm may reorder the whitelists of the radios links that are scheduled at the same timeslot in order to eliminate internal collisions while still multiplexing the different transmissions through different radio channels for mutually interfering links. Thus, the radio links use in their transmissions the radio channels that present high-reliability in their propagation range (channels' spatial diversity).

Finally, we proposed an ***hybrid*** blacklisting technique, called Adaptive MABO (AMABO). MABO-TSCH [84] is a well known blacklisting technique, which assigns a collection of channel offsets for each link so that the schedule may be computed centrally, while the blacklist may be constructed locally. However, the MABO-TSCH approach is not adaptive, since the same amount of radio resources is allocated to each radio link. Therefore, we proposed AMABO, that assigns the multiple channels offsets to the radio links dynamically in per timeslot basis. AMABO's technique achieves to exploit the full range of the available channels at any given time, thus increasing the number of channels assigned per radio links. As a consequence, it increases the probability to use a good channels for the data transmissions.

Chapter 4

Efficient Resource Allocation in TSCH-based Wireless Mesh Networks

In the previous chapter, we explored how blacklisting techniques may improve the end-to-end network reliability. However, a low-power wireless mesh network, in addition to high reliability, has to provide bounded latency. Unfortunately, guaranteeing a bounded end-to-end latency is particularly challenging since transmissions have to be in temporal proximity. Even worse, a potential link quality degradation may result in reconstructing the whole TSCH-based schedule along the path. Therefore, an appropriate organisation of the transmissions (and receptions) is required to bound the end-to-end latency, i.e., a scheduling algorithm.

As it was presented in the Introduction Chapter, the scheduling algorithms can be classified as *centralized*, *distributed*, and *autonomous*.

The 6TiSCH WG proposes a reactive and distributed scheduling solution known as the MSF [33]. This scheduling function defines the bootstrapping process for a node to join the network, and how the communication schedule is managed in a distributed fashion. In this Chapter, we first present our exhaustive works on performance evaluation of the MSF [34], [62] which demonstrate that the standardised scheduling function is not sufficiently robust. Then, we present our state-of-the-art contribution, called Low-latency Distributed Scheduling Function (LDSF) [29], tailored to provide both high-reliability and bounded end-to-end latency. LDSF is a fully distributed algorithm, where the constrained devices in the network decide by themselves the cells to use. LDSF relies on the organisation of the TSCH slotframe in smaller parts, called blocks. The thorough performance evaluation demonstrates that LDSF outperforms most of the state-of-the-art distributed scheduling functions, the MSF [33], LLSF [56] and Stratum [57]. Even though LDSF is one of my contributions that I am very proud of, it is based on a distributed approach. In industrial environments, high network reliability and low latency performance are not sufficient. Indeed, in control loop scenarios, several consecutive data packet losses are unacceptable, while if a data packet arrives to the destination after the pre-determined time, then this data packet is considered lost. There-

fore, we then worked on centralised approaches, and proposed a solution that allows very high network reliability even under real-world conditions such as the Battery Management System (BMS) in an Electric Vehicle (EV) [21], [88].

These works have been conducted during the ***PhDs of Vasileios Kotsiou and Guillaume Le Gall***, whom I had the honour to supervise, the last year of the ***Postdoctoral period of Remous-Aris Koutsiamanis*** as well as during the last year of the ***PhD of David Hauweele*** (University of Mons, Belgium) with whom I had the opportunity to collaborate with.

4.1 Thorough Performance Evaluation of 6TiSCH Minimal Scheduling Function (MSF)

The starting point for scheduling the resources of a 6TiSCH network is the MSF and it's use of the 6P protocol to negotiate resource allocation. More specifically, the 6P protocol defined in 6TiSCH WG only provides the necessary transactions to manipulate cells in each node's schedule. It is up to the scheduling function to decide when to add or delete cells from those schedules by triggering 6P request/response commands. To this end, a number of SFs has been proposed in the literature to optimise the resource allocation process in a 6TiSCH-likenetwork [35]. Furthermore, the 6TiSCH WG defined a *distributed* scheduling solution known as the MSF [33], which allows the negotiation and reservation of network resources (i.e., cells) in an on-demand manner. This scheduling function defines the bootstrapping process for a node to join the network, and a subsequent mechanism for each node to adapt to traffic changes, routing changes, and schedule collisions.

Types of Cells

MSF relies on 3 different types of cells for its operations:

- The **Minimal cell** which is a single mandatory shared cell (located at timeslot 0 and channel offset 0) used to bootstrap the network [32], and ensure minimal connectivity between the nodes in the network.
- MSF also defines **Autonomous cells** that act as default rendez-vous points to bootstrap unicast communications with their neighbours. These cells are called "autonomous cells", because they are maintained autonomously by each node without negotiation through 6P.
- Finally, MSF allocates **Negotiated cells** that will be used by a node for communication and announcing itself to potential newcomers. Such cells are negotiated by a node with its neighbours through 6P transactions, according to the current traffic load.

Network Bootstrapping

A node expecting to join a 6TiSCH network must go through a series of steps before being able to transmit messages within the network. First, it must discover and synchronise with the network. Then, it must learn keying material and setup routing to its preferred parent. Finally, it must negotiate cells. This process is divided into 6 steps as detailed in [33] (Section 4).

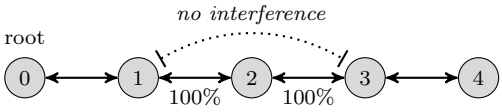


Fig. 4.1: Linear topology with a link quality of 100% between adjacent nodes, and no interference between non-adjacent ones. *Taken from* [62].

Addition / Deletion Rules

MSF dynamically adapts the number of negotiated cells of each node. This happens in the three following cases. Firstly, when the available link-layer resources are adapted to the current traffic load. Secondly, when a new preferred parent is selected, as part of RPL operations and cells must be re-negotiated. Finally, when certain cells experiencing excessive schedule collisions need to be relocated.

In the case of traffic load changes, a node adapts its number of negotiated cells when it detects a significant increase or decrease in traffic. To this end, it estimates the traffic load over a recent window of time expressed as a number of cells. This is done by maintaining a pair of counters (*NumCellsPassed* and *NumCellsUsed*) per neighbour and per traffic direction. *NumCellsPassed* counts the elapsed number of scheduled cells to the preferred parent whether or not they resulted in a transmission, while *NumCellsUsed* counts the subset of those cells that *were used* for a transmission, whether or not that transmission was successful.

A node updates and adapts its schedule after a certain number of cells, *MAX_NUMCELLS*, has passed.

Performance Evaluation

Simulation Setup

We employed the discrete-event 6TiSCH simulator [89], which implements a careful abstraction of the 6TiSCH stack, to perform a thorough evaluation of MSF. First, we evaluated it over a constant traffic, then with varying traffic to assess the adaptation features of MSF. We performed these evaluations over a linear topology, see Fig. 4.1. This linear topology that comes with "perfect" link qualities and with no interference between each pair of adjacent nodes, allowed us to investigate MSF at a fundamental level, i.e., cell allocation. Finally, we only considered traffic going upstream, through the preferred parent based on RPL routing protocol. The detailed configuration of the evaluated algorithms is presented in [62], [34].

Simulation Results

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [34], [62] the detailed performance evaluation results are demonstrated.

Constant Traffic: First, we focused on constant traffic, where each node from 1 to 4 generates traffic with rate R ranging from 0.1 up to 10pkts/sf.

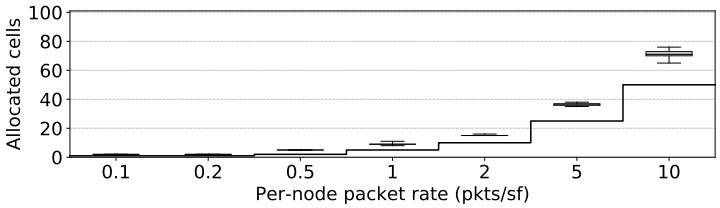


Fig. 4.2: Number of cells allocated on node 2, as a function of per-node packet rate. The boxes represent the amount of cells allocated across multiple runs of the same simulation. The black line represents the minimum amount of cells, N_{req}, required to transport the CBR traffic. *Taken from* [62].

In Fig. 4.2, for each traffic rate, the number of cells allocated by MSF on node 2, together with the theoretical number of cells required (line steps) is shown. The theoretical number of cells on node 2 is obtained as $N_{\text{req}} = \lceil 5 \times R \rceil$, where R is the per-node traffic rate. The factor 5 comes from the fact that node 2 receives data packets from node 3 and 4 and forwards them upstream along with its own packets. Let's consider the case of $R = 5$pkts/sf. The theoretical number of cells required is 10 RX and 15 TX cells, for a total of $N_{\text{req}} = 25$ cells, while the median (resp. maximum) number of allocated cells in our experiments is 36 (resp. 38). Note that over-provisioning was expected because additional cell allocation will stop only when the MSF TX estimator falls below the high cell usage threshold, that is when MSF TX $\leq \frac{LIM\_NUMCELLSUSED\_HIGH}{MAX\_NUMCELLS} = 75\%$. The theoretical number of cells, taking into account over-provisioning, can be estimated by Eq. 4.1. For $R = 5$, that gives $N_{\text{ovp}} \approx 33$, which is close to the observed results.

$$N_{\text{ovp}} = \frac{MAX\_NUMCELLS}{LIM\_NUMCELLSUSED\_HIGH} \times N_{\text{req}} \tag{4.1}$$

Varying Traffic: Next, we focused on MSF's ability to allocate or deallocate resources when the traffic load changes. To do so, we used a simpler setup with only two nodes, the root and one leaf node sending traffic at a packet rate that periodically changes. More specifically, every 500 seconds, the sending application cycles through the following rates: 10, 20, 30, 20, 10 and finally back to 0pkts/sf. We measured the time required from the moment the packet rate changed to the moment it reached a stable schedule in the slotframe.

Fig. 4.3 shows the evolution of several parameters along time for a single run of this simulation. This figure is split into three parts. The middle one shows the evolution of the transmit queue length (TxQ) and the MSF estimation of the traffic load (MSF TX). The bottom part shows the evolution of the number of allocated cells along with the theoretical minimum number of cells. The top part shows when MSF decides to allocate new cells (up arrow) or to deallocate existing cells (down arrow).

The simulation starts at $t = 0$. The traffic rate suddenly increases from 0 to 10pkt/sf and as a consequence, TxQ jumps to 100% occupancy as there are insufficient cells. MSF is activated and slowly allocates new cells through 6P ADD requests. The rate at which new cells are allocated rapidly increases as it takes less and less time for a period of *MAX_NUMCELLS* to pass. At $t = 316$s, MSF has converged to a stable state, the slotframe contains enough cells to carry the traffic load. At $t = 500$s, the traffic rate jumps from 10 to 20pkts/sf,

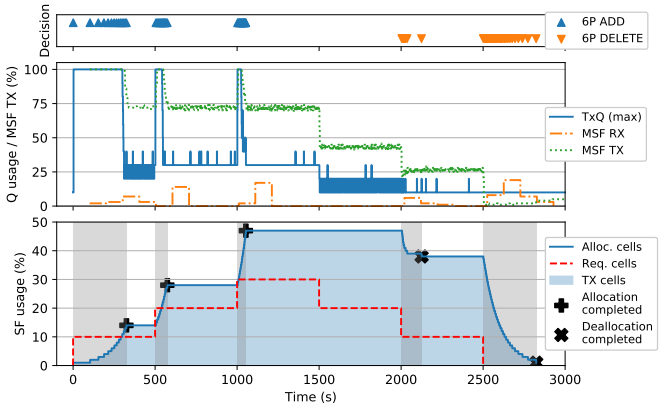


Fig. 4.3: Evolution of allocated cells along time with a traffic load varying in the 0 – 30pkt/sf range with rate change steps of 10pkt/sf each. Allocation/deallocation periods are shaded in gray. *Taken from* [62].

leading to another round of cell allocations that ends at $t = 565$s. Although this jump in traffic rate is equal in intensity to the first one, the time to adapt was much shorter. At $t = 1000$s, the last increase in traffic rate takes place, jumping from 20 to 30pkts/sf. It requires an even shorter convergence time (50s).

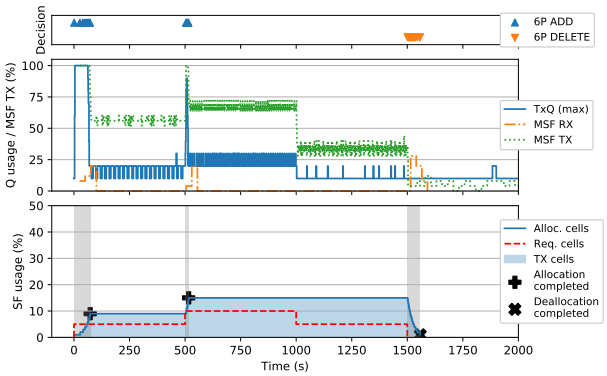
After $t = 1500$s, the traffic decreases from 30 to 20pkts/sf. However, MSF withholds the decision to deallocate cells as MSF TX does not drop below the 25% limit, *which results in a high over-provisioning level*. At $t = 2000$s, the traffic decreases again from 20 to 10pkts/sf. This time, MSF triggers deallocations but only for a handful of cells until it reaches the lower limit of 25%. After $t = 2500$s, the traffic drops back to 0pkt/sf resulting in a value of MSF TX of $\approx 0\%$. Hence, MSF deallocates all but one cell during a period of 279s.

Towards faster resource allocation: We studied the impact of changing implementation constants on the convergence pattern of the scheduling function. Indeed, to accelerate the resource allocation, the MAX_NUMCELLS constant, which represents the length (in number of elapsed cells) of the window used to estimate the current traffic load, could be reduced [90]. Reducing the size of this window would reduce the time between a change in traffic load and the decision to allocate more or free existing resources, thus speeding up the convergence.

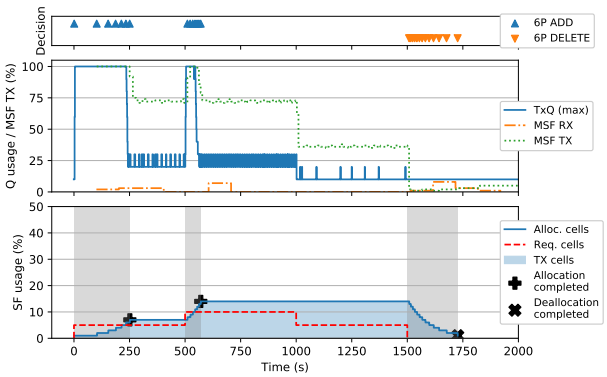
However, reducing *MAX_NUMCELLS* has a side effect: it makes the estimation of the current traffic load less precise as it is computed on a shorter sample. Moreover, the granularity of the estimation is given by $\frac{1}{MAX\_NUMCELLS}$. A coarser granularity and a less accurate estimation of the load can trigger unexpected 6P transactions, and even oscillations. To further investigate the effect of *MAX_NUMCELLS* on the behavior of MSF, we reproduced the simulations with dynamic traffic. However, in this set of experiments, we increased the traffic rate from 0pkt/sf to 5pkts/sf, then to 10pkts/sf, then we reduced to 5pkts/sf, and finally back to 0pkt/sf.

As expected, the convergence time is almost directly proportional to *MAX_NUMCELLS*. Indeed, switching from a window of size 100 (Fig. 4.4b) to 25 (Fig. 4.4a) reduces the time to allocate the resources from 250.46s to 71.69s during the first convergence period. On the contrary, by increasing the *MAX_NUMCELLS* to 200 (Fig. 4.4c) extends the convergence time of this period to 497.91s. Furthermore, even though the traffic is purely periodic with perfect link quality (PDR=100%),

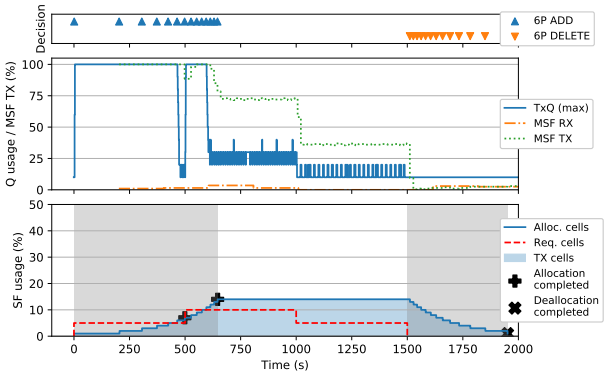
by reducing the *MAX_NUMCELLS*, the variance of the MSF TX estimator increases. If that pattern is recurring, then the estimation will oscillate with a value above the high 75% threshold and a value below the low 25% threshold, and, thus, potentially triggering constant allocation and deallocation.



(a) MAX_NUMCELLS = 25.



(b) MAX_NUMCELLS = 100 (default).



(c) MAX_NUMCELLS = 200.

Fig. 4.4: Evolution of allocated cells for different values of *MAX_NUMCELLS*. Large values of *MAX_NUMCELLS* result in a long convergence time. Lower values of *MAX_NUMCELLS* increase the variance on MSF TX. *Taken from* [62].

4.2 Distributed Approach: Low-latency Distributed Scheduling Function (LDSF)

As presented in the previous section, our exhaustive study on performance evaluation of the MSF [34], [62] demonstrates that the standardised scheduling function is not sufficiently robust. Therefore, we proposed the Low-latency Distributed Scheduling Function (LDSF) in [29], tailored to provide both high-reliability and a low end-to-end latency. LDSF is a fully distributed scheme, where each device in the path decides by itself the cells to use. The proposed solution relies on the organisation of the slotframe in smaller parts, called blocks. Each transmitter selects the right set of blocks, depending on its hop distance from the root, so that retransmission opportunities are automatically scheduled. Therefore, the impact on the end-to-end delay when the packet has to be retransmitted by a transmitter is limited. The transmission opportunities are still chained further chained to limit the buffering delay.

Slotframe Organisation

In this work, we considered sporadic traffic, where each sensor reports periodically its measurements to a border router. Thus, the slotframe length has to be equal to the least common multiplier of all the traffic periods. Consequently, in each long slotframe, shared cells are reserved for control traffic, such as 6P packets. One dedicated cell is also reserved for each node to send its unicast control packets corresponding to routing or synchronisation. All data packets are transmitted through dedicated cells, that each pair of nodes has to reserve.

We proposed to organise the long slotframe into small blocks that repeat over time.

To reduce the end-to-end delay, we limited the *buffering delay* when a packet is retransmitted. By reserving consecutive blocks for retransmissions, the buffering delay is proportional to the block size. Thus, we divided the slotframe into small blocks with a few timeslots.

A packet is typically received during a block, and forwarded during the next one. Thus, a transmitter selects its block according to its hop count from the border router. More precisely, each block has a block id, that counts the number of blocks since the beginning of the slotframe. We have consequently even blocks (with an even block id), and odd blocks (with an odd block id). A transmitter has to select a block, so that the remainder of the Euclidean divisions of the hop count and of the block id by 2 are equal. More formally, a transmitter can select any cell in the blocks which respect the following property:

$$HC \pmod 2 = B_{id} \pmod 2 \tag{4.2}$$

where HC denotes the hop count from the transmitter to the border router, and B_{id} the block id.

Let us consider the topology and the LDSF schedule illustrated in Fig. 4.5. The node A is two hops away from the border router, and the packet is assumed to be generated in the timeslot 0. It must select a block with an even block id ($2 \pmod 2 = 0$). In our example, it selects the timeslot 2. The node C is 1 hop away from the border router, and considers only the blocks with an odd block id. It selects the block 1 (the consecutive block), and reserves one cell (here,

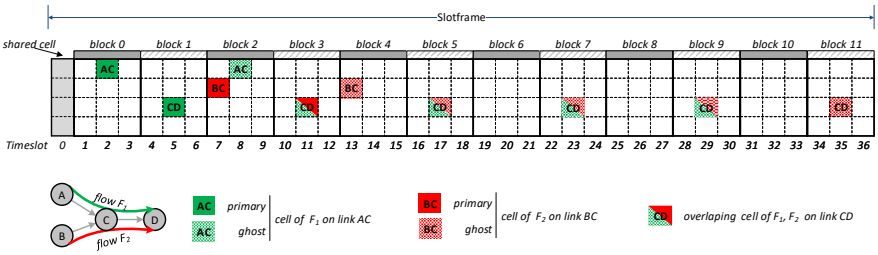


Fig. 4.5: Slotframe organization in blocks, where $MaxRetries = 1$. *Taken from* [29].

the timeslot 5) to forward the packets from A. The blocks are daisy-chained, the packet received during the block i being forwarded in the block $i + 1$.

In this work, we made a distinction between the following types of cells:

- **Shared cell** for control packets in broadcast (EBs, routing advertisements), and control packets in unicast when no dedicated cell has been reserved (i.e., 6P requests/replies);
- **Primary** (dedicated) **cell** corresponds to the earliest expected reception time of the data packet from the previous hop (or from the application layer);
- **Ghost** (dedicated) **cells** correspond to the retransmission opportunities, that are automatically used by the transmitter if it did not receive an acknowledgment for its previous transmission. The ghost cells are scheduled in the same timeslot and channel offset as for the corresponding primary cell, but in the subsequent blocks.

When a node reserves a primary cell in a block, a fixed number of ghost cells is automatically reserved every two blocks. Thus, we can daisy chain the transmission opportunities along the path: a node is able to receive a packet during a block, and forward it during the subsequent blocks. In this way, we maintained a low end-to-end delay.

A link quality degradation means more retransmissions: in classical scheduling algorithms, we would need to reserve additional cells. In LDSF, a large number of cells is pre-reserved, at the very beginning, to cope with the worst link qualities. Thus, the number of ghost cells (for retransmissions) is fixed, whatever the link quality. Besides, the impact of the retransmissions on the end-to-end delay is limited since the blocks comprise a small number of timeslots. Note that the ARQ feature of TSCH was employed, where the transmitter schedules a retransmission in the next ghost cell only if it does not receive an acknowledgement for its previous transmission.

Number of Ghost Cells

In the following, the computation of the number of ghost cells provisioned for the retransmissions is presented.

Standard case

The delay induced by the retransmissions is cumulative along the path. Thus, we had to cope with the worst case: a packet may be retransmitted at most $MaxRetries$ times by each transmitter in the path. The latest time of arrival corresponds to the last RX ghost cell (e.g., C receives the packet from A at the latest during the timeslot 8 in Fig. 4.5).

We made a distinction between the *source* of the flow that generates a data packet, and a *transmitter* that forwards this packet. The first transmitter in the path corresponds trivially to the source.

Note that the number of ghost cells is proportional to the hop distance from the source. More precisely, a transmitter has to provision $(MaxRetries * (Hops + 1))$ ghost cells for the retransmissions, where $Hops$ denotes the hop distance from the source to the transmitter.

In Fig. 4.5, A is the source ($Hops = 0$) and provisions one primary cell (timeslot 2) and one ghost cell (timeslot 8). For the node C, it is one hop away from the source ($Hops = 1$). Thus, C allocates for the flow F_1 one primary cell (timeslot 5) and 2 ghost cells ($MaxRetries * (Hops + 1)$). We can note that the node C can receive the packet through the primary or the ghost cells. Thus, even if it receives a packet during the last ghost cell (timeslot 8), it has still two transmission opportunities (timeslots 11 and 17) for one transmission, and one retransmission.

Overlapping case

Some flows may overlap, i.e., one relaying node uses the same ghost cells for two different flows. For instance, flows F_1 and F_2 are both forwarded by the node C, where some ghost cells are in common for both flows. A node can detect an overlap when receiving a 6P request: the primary cell corresponds to a ghost cell already reserved for another flow.

Even with this overlap, enough ghost cells have to be guaranteed to handle the worst case. Let us consider the two following cases:

Case 1) the node receives a packet from the novel flow (F_2) while a packet from the previous flow (F_1) is already in the queue. By construction, the first flow F_1 has still enough ghost cells to handle $MaxRetries$ retransmissions. At the latest, the packet for the flow F_2 is received while only $MaxRetries + 1$ ghost cells remain in its schedule (primary transmission + retransmissions). Thus, $MaxRetries + 1$ ghost cells for the novel flow F_2 should be provisioned, after the ghost cells that would have been allocated to the flow F_2.

Case 2) the node receives a packet from the novel flow (F_2) while the packet from the other flow (F_1) was not yet received. For the same reason, the node has enough ghost cells for F_2 for $MaxRetries$ retransmissions. Thus, we included $MaxRetries + 1$ additional ghost cells at the end of the range, but they will be used to forward the packet for the flow F_1.

To conclude, it is sufficient to provision $MaxRetries + 1$ additional ghost cells when an overlap is detected, whatever the hop distance from the source.

Scheduling process

In LDSF, shared cells are only used for signalling, i.e., sending/receiving the 6P packets to negotiate which dedicated cells to use. A 6P request typically

piggybacks a list of possible (dedicated) primary cells. The (dedicated) ghost cells are automatically derived from a primary cell. The receiver sends a 6P reply to the transmitter to validate the reservation. Since no dedicated cell is present in the schedule, the 6P packets use the shared cell.

A novel allocation is triggered when a node receives either a 6P request from the previous hop or directly the packet from the application layer. The detailed scheduling process is presented in [29], see Section IV.C.

Energy Savings using Ghost Cells

Reserving ghost cells allows the network to improve the reliability, as LDSF can efficiently handle the fast link quality changes since ghost cells are a priori over-provisioned. Concerning the energy efficiency, the transmitter can safely sleep when it has no data packet to transmit. For the receiver side, we have to limit idle listening [91], forcing the node to wake-up at the beginning of the timeslot because it is not aware that the transmitter has nothing to transmit.

Under the LDSF algorithm, we configured a fixed number of ghost cells, based on the hop distance from the source, and a constant, whatever the link quality. A receiving node must wake-up at the beginning of each primary cell, to possibly receive a packet. Then, it must also wake-up for all the subsequent ghost cells until a packet has been received and correctly acknowledged. Once, a packet has been received or the last ghost cell is encountered, the receiver can safely save energy until the next primary cell. The receiver has then to forward the packet, and becomes a transmitter. It selects the corresponding cell in the next blocks, and starts to transmit the packet to the next hop.

Let us consider the scenario illustrated in Fig. 4.5. Let us assume that the node C has been able to decode the packet from the node A in the timeslot 2. It can stop listening to the ghost cell in timeslot 8. However, it will wake-up during the next block (timeslot 5) to forward the packet to the node D.

The primary cell corresponds to the earliest time arrival to optimise the end-to-end delay. Thus, we do not have any false negatives: the receiver is always awake when the transmission takes place, we thus keep the deterministic behaviour of TSCH.

Performance Evaluation

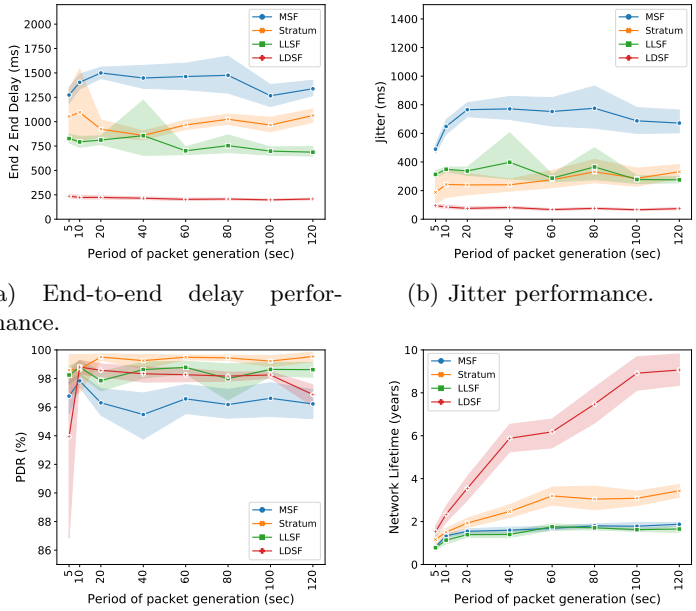
Simulation Setup

In order to evaluate the performance of our proposed distributed scheduling function, we employed the 6TiSCH Simulator [89], a discrete-event simulator written in Python. We generated random topologies, where each node is randomly located in an area of $2000 \times 2000m^2$. Thus, each node has at least 3 neighbours. The propagation model of 6TiSCH Simulator is based on the Pister-Hack model [92].

Scheduling Algorithms

In addition to MSF, we extended the 6TiSCH Simulator with three following scheduling functions (LDSF, LLSF [93] and Stratum [94]):

MSF: [33] the standardised 6TiSCH scheduling function.



(a) End-to-end delay performance.

(b) Jitter performance.

(c) End-to-end PDR.

(d) Network lifetime.

Fig. 4.6: Impact of the traffic rate (i.e., inter packet time). *Taken from* [29].

Stratum: [94] divides the slotframe in blocks (i.e., stratums). Each node selects a block according to its hop distance.

LLSF: [93] aims to reduce the end-to-end latency by allocating receiving and transmitting cells as close as possible in the schedule.

LDSF: Our proposed scheduling function.

Stratum uses a slotframe length of 101 timeslots, to be able to provide an end-to-end delay equal to $1010ms$ (=101 * $10ms$). MSF and LLSF also uses the default slotframe length (101). LDSF uses rather a slotframe length proportional to the maximum flow rate, since it was designed for this purpose. The same cell is used every two blocks for transmissions and retransmissions. Since each block comprises 5 timeslots in LDSF, the transmitter has to wait on average $10ms$ * 5 timeslots * 2blocks = $100ms$.

Our implementation (simulation code, scripts, and raw data) is freely available (`https://github.com/vkotsiou/Scheduling` for the implementation, and `https://doi.org/10.5281/zenodo.3748712` for our dataset).

The detailed configuration of the evaluated algorithms is presented in [29].

Simulation Results

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [29] the detailed performance evaluation results are demonstrated.

We first measured the average end-to-end delay (Fig. 4.6a). MSF presents the highest end-to-end delay, because it does not have any cell allocation strategy to minimize the delay, as it selects randomly the cells. LLSF and Stratum

presents somehow stable performance, however, still they come with very high numbers. Under LLSF, while the cells are reserved consecutively along the path, the first cell is picked randomly and, thus, generates a large buffering delay (half of the slotframe = 505ms). Finally, LDSF comes with the best delay performance, and it is very robust to large traffic rates.

Next, as it can be observed in Fig. 4.6b, the jitter performance follows the one of the delay. MSF provides the worst jitter since retransmission cells can introduce a cumulative effect along the path, since they can be allocated after the cells of the next hop. LLSF and Stratum achieve similar jitter performances. Under Stratum, a packet is randomly scheduled in the last *stratum* (i.e., block) to be received by the border router. Since this *stratum* is typically much larger than the LDSF's block, the jitter is mechanically increased. LDSF provides the lowest jitter performance which is less than $150ms$, even for very high traffic rates. Collisions are accurately handled, and the packets are retransmitted efficiently in the subsequent blocks to minimize the buffering delay.

In Fig. 4.6c, the reliability performance is depicted. All schemes are able to guarantee end-to-end reliability above 96% in most cases. Stratum achieves the highest reliability for low traffic rates since the blocks are large to avoid collisions. However, the number of collisions starts to increase for high traffic rates (inter packet time $< 10s$). LDSF provides an end-to-end packet delivery ratio above 98%. LLSF provides also a good reliability, except for high-traffic rates: many collisions arise and are particularly challenging to resolve since the cells are contiguous. Moreover, the scheduling process needs to solve the collisions for each cell, while LDSF is more robust since the same cell is pre-reserved also for the retransmissions.

Finally, in Fig. 4.6d, the network lifetime is illustrated. We extrapolated the average energy consumption for the most loaded node to derive the network lifetime. MSF generates a large number of control packets with many (de)allocations, which impact negatively the lifetime. Stratum increases slightly the lifetime, by reducing the renegotiation of cells. LLSF achieves similar performance, since it uses a short slotframe where the shared cells consume energy. Finally, LDSF efficiently handles the lossy radio links: ghost cells are automatically reserved after 6P transactions, minimizing the amount of control traffic.

4.3 Centralised Scheduling Functions:

The BMS of an EV is a system designed to ensure safe operation of the battery pack, and reporting its state to other systems. In today's BMS implementations, the communication is performed through wire buses. Recently, there have been several initiatives toward wireless communication in a BMS application [21], [88], [95]. A battery pack is divided into modules made of a series. Each module is supervised by a board named Cell Sensor Unit (CSU), which communicates with a Master Control Unit (MCU) to report their data. In a BMS, the number these cells is relatively small: it could be as high as 96 if there is one CSU per cell, but a network of 16 nodes is probably a more realistic number (when 6 cells per CSU is considered). Moreover, the number of nodes is fixed and is known in advance, and all nodes can have direct communication with each other. Therefore, in such scenarios a centralised solution is more suitable both for topology management and scheduling.

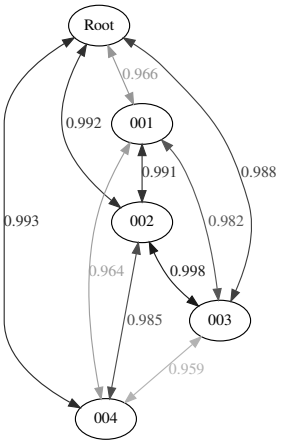
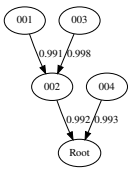
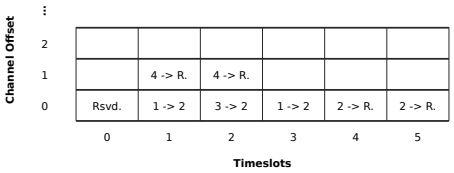


Fig. 4.7: Example problem graphical description with 4 nodes. *Taken from* [88].



(a) Output topology.



(b) Output schedule.

Fig. 4.8: Output topology and schedule for the 4 nodes network example. *Taken from* [88].

We first proposed a centralised network management technique for uplink very reliable periodic traffic (every $250ms$) based on Linear Programming (LP) [96], which is a well-known approach for obtaining an optimal solution to a problem, by maximising (or minimising) an OF, under several constraints. We then proposed an improved version of this technique by using the Simple Descent (SD) based approach [97]. Our performance evaluation campaign demonstrated that SD strategy is more efficient in terms of processing time.

LP-based BMS Wireless Network

We proposed a two-steps process to build a topology that uses the best possible links, and then a TSCH-based centralised schedule. Both decision process steps employed LP. The objective was to allocate network resources (radio links and uplink TSCH cells) to provide reliable data delivery (the voltage and temperature data collection) of all nodes before within a slotframe boundaries. To solve this topology management and scheduling problem, we implemented an application in C++ that takes all the possible links between the nodes with their PDR as input, and print the resulting schedule and topology as output.

Building the Topology: In Fig. 4.7, an example input with randomly generated link qualities is illustrated. The random values have been chosen in the [0.95; 1] interval, to resemble the experimental values that have been measured in [21], [88]. For the sake of simplicity, we assumed the links are

symmetrical in terms of quality. In this example problem, by employing the LP method, we opted to maximise the average path PDR for each node.

In the following, the variables of the problem are designated with two indexes. The first one is the ID of the transmitter, and the second one is the ID of the receiver. We labeled N the number of non-root nodes, which gives a total number of nodes, including the root, of $N+1$. The variables can be represented as in Equation (4.3).

$$X_{i,j}\ i \in [1;N], j \in [0;N] \tag{4.3}$$

In this equation, if the link is used, with i having j as parent, the variable is equal to 1, and it is equal to 0 otherwise. We used the δ variable as follows:

$$\delta_{i,j} = \begin{cases} 0 & if\ i = j \\ 1 & otherwise \end{cases} \tag{4.4}$$

The OF to be maximized is presented in (4.5).

$$F = \sum_{i=1}^{N} \sum_{j=0}^{N} P(i,j) X_{i,j} \tag{4.5}$$

with:

$$P(i,j) = \begin{cases} {p_{i,0}}^2 & if\ j = 0 \\ K\ p_{i,j}\ p_{j,0} & otherwise \end{cases}$$
$$K = 1.1$$

We decided to maximise this OF under the following constraints:

- Each non-root node has a maximum of A children.

$$\forall i \in [1;N], \sum_{j=1}^{N} \delta_{i,j} X_{i,j} \leq A \tag{4.6}$$

- Each node has one parent.

$$\forall i \in [1;N], \sum_{j=0}^{N} \delta_{i,j} X_{i,j} = 1 \tag{4.7}$$

- Each node has a path to the root.

$$\forall i \in [1;N],\ \forall j \in [1;N],\ i \neq j,$$
$$X_{i,j} + \sum_{l=1}^{N} \delta_{j,l} X_{j,l} \leq 1 \tag{4.8}$$

The presented program ran with the data shown in Fig. 4.7 as input, and it gives the topology result shown in Fig. 4.8a as output.

Building a TSCH Schedule: At this stage of the network management process, the software takes the result that was obtained by the **Building the**

Topology feature as input, and outputs the resulting TSCH schedule. It is assumed that the application data payload is small enough to be aggregated at the intermediate nodes, and the resulting packet is small enough to be sent within one timeslot to the root node.

The variables are designated with two indexes. The first one is the ID of the transmitter node, and the second one is the timeslot offset in the slotframe, starting at 0. We do not need to specify the receiver node, since it has been determined in the previous step. We label S the last timeslot index that may be used. The variables can be represented as in (4.9).

$$X_{i,k} \; i \in [1; N], k \in [0; S] \tag{4.9}$$

Furthermore, we labeled R_1 the set of node IDs that have a rank of 1 in the topology, i.e., which have the Root node as parent. We label R_2 the set of node IDs that have a node in R_1 as parent. We label $C_i, i \in [1; N]$ the children nodes of i, that are of rank 2. Note that the generated schedule (slotframe) always has a first timeslot which is marked as Reserved and which should be used for beacon frames and control plane traffic.

In this problem, we opted to minimise the TSCH slotframe size, while keeping the path PDR of every node above a given threshold. Indeed, by allocating all the timeslots of the nodes with high rank values (nodes that are far from the root) to the most left available timeslots, then allocating the timeslots for all the nodes of *Rank 1* to the most left available timeslots within the slotframe, while taking into account the following constraints.

- If a node is sending its data packet through a relay node to the root, it must send it to the relay node before that node sends the aggregated data packet to the root. In other words, we impose a happens-before relationship between receiving and sending for the R_1 nodes (4.10).

$$\begin{gathered} \forall i \in R_1, \; \forall j \in C_i, \forall k_1 \in [0; S], \forall k_2 \in [k_1; S], \\ X_{i,k_1} + X_{j,k_2} \leq 1 \end{gathered} \tag{4.10}$$

- The channel offset is limited to C channels (4.11).

$$\forall k \in [0; S], \sum_{i=1}^{N} X_{i,k} \leq C \tag{4.11}$$

- Each node can only use its radio for one dedicated timeslot in a same timeslot. It is sufficient to input this constraint for nodes in R_1 (4.12) and for the root (4.13).

$$\forall i \in R_1, \; \forall k \in [0; S], \; X_{i,k} + \sum_{j \in C_i} X_{j,k} \leq 1 \tag{4.12}$$

$$\forall k \in [0; S], \; \sum_{i \in R_1} X_{i,k} \leq 1 \tag{4.13}$$

- Each node must have at least one transmit timeslot scheduled (4.14).

$$\forall i \in [1; N] \sum_{i=1}^{N} X_{i,k} \geq 1 \tag{4.14}$$

- The path to the Root for each node must have a PDR above the given threshold (4.15).

$$\forall i \in [1; N] \sum_{k=0}^{S} X_{i,k} \geq T_i \tag{4.15}$$

  where T_i is the minimum number of transmissions for the link with sender node i to meet the minimum path PDR threshold requirement.

In equation (4.15), to calculate the number of transmission timeslots to be allocated for each link, we consider the PDR without retransmission of each link, and the minimum path PDR. To do this, we iteratively schedule a retransmission on the worst link of the worst path, until all the path PDRs are above the requested threshold. After the retransmissions have been scheduled, the resulting link PDR is as shown in (4.16).

$$PDR_{rtr} = 1 - (1 - PDR)^{tr} \tag{4.16}$$

In this equation, PDR_{rtr} is the PDR after all the transmissions have been scheduled, and tr is the total number of transmissions for that link.

The LP program is described in [98], see section 4.2. It ran with the data shown in Fig. 4.8a as input gives the result shown in Fig. 4.8b as output.

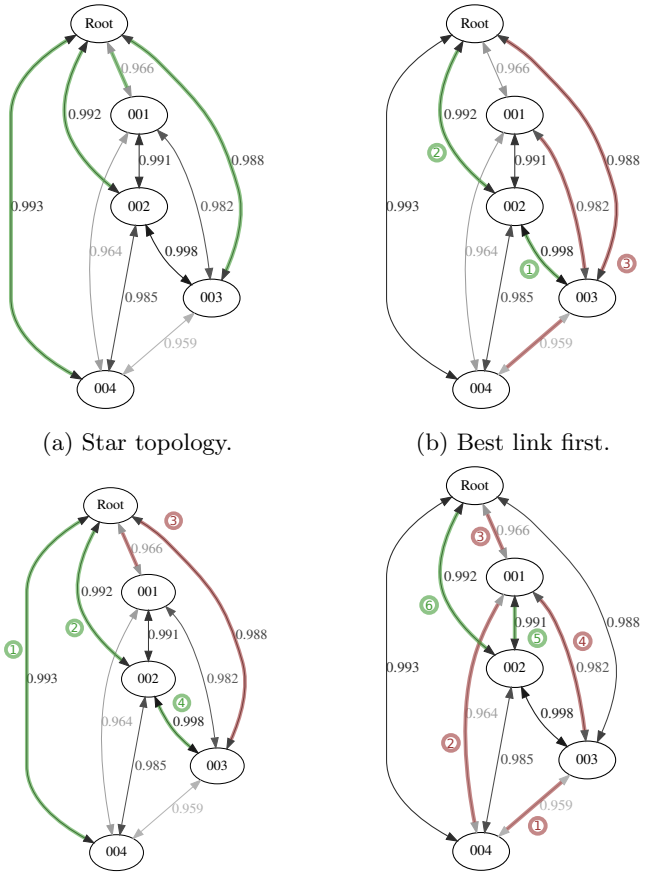
SD-based BMS Wireless Network

The LP strategy gives an optimal result under the selected constraints, but computing the topology may take time in some scenarios, which is a well-known downside of LP. Therefore, we opted for the SD method to build the topology, while for the optimal schedule, we used a *procedural* approach, by allocating all the timeslots of the nodes with R_2 to the most left available timeslots, then allocating the timeslots for all the nodes of R_1 to the most left available timeslots, while taking into account the constraints exposed in **Building a TSCH Schedule** section.

In the SD method, the goal is to rapidly find an approximate solution to the problem. This is done by first building a valid solution, and then applying permutations to it to keep this intermediate result a valid solution at all iterations, while converging towards a local minimum or maximum, according to the OF.

To build an initial solution, we used four methods, described below. In the associated examples, the green colour means the radio link has been selected, and the red colour means it has been removed from the available radio links list. The numbers reflect the order in which these operations are performed.

- **Star topology**: All the nodes have a direct radio link to the root. An example is available in Fig. 4.9a.



(a) Star topology.

(b) Best link first.

(c) Best link to root first.

(d) Eliminate weakest link first.

Fig. 4.9: The 4 proposed methods to build an initial solution. An example of a network that consists of 4 nodes plus the Root. *Taken from* [88].

- **Best link first**: This is an iterative process in which the best radio link is selected. If it is a radio link to a non-root node, then this intermediate node has its radio link to the root selected, while the other possible radio links to the now R_2 node are removed from the available radio link list Fig. 4.9b.
- **Best link to root first**: This is an iterative process in which the best link to the root are selected first, until the root has the minimum possible child nodes. The remaining links to the root are removed from the link list. Then, the remaining nodes, which are then of R_2, have their link selected using the *"best link first"* method, see Fig. 4.9c.
- **Eliminate weakest link first**: This is an iterative process in which the worst links are eliminated. Every time a link is removed from the link list, the algorithm checks if for each node they have more than one path to the root. When a node has only one possible path left, these links are selected, see Fig. 4.9d.

In the second part of this process, we applied permutations to the current solution and calculated the value of the OF defined in 4.5 for each permutation

that has been tried. The permutation with the highest OF value becomes the current solution for the next iteration. When no permutation gives a better result than the current solution, a local maximum has been reached and the current solution then becomes the topology. The following permutations have been used to try to approach the optimal solution:

- **Exchange a R_1 node with a R_2 node**: switch the position of a R_1 node with a R_2 node in the topology.
- **Exchange two R_2 nodes**: switch the position of two R_2 nodes in the topology.
- **Make a R_1 node a child of another R_1 node**: select a node with R_1 with no child, and make it R_2 as child of another R_1 node.

Note that the resulting topology obtained with this method may vary depending on the initial solution chosen and the set of permutations used.

Performance Evaluation

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [21], [88], [98] the detailed performance evaluation results are demonstrated.

Choosing the initial solution

Considering that aggregation of data packets may happen at R_1 nodes, the most meaningful approach to evaluate the proposed solutions is probably the average weighted link PDR, as in this case the PDR for a link will be counted as many times as there will be data packets in the aggregated packet. The four initial solution building methods have been tested under scenarios with random radio link qualities in the $[0.95; 1]$ interval, for a number of nodes varying from 2 to 32, and 100 times for each number of nodes. The detailed simulation setup is presented in [88].

The results are illustrated in Fig. 4.10. The best results are obtained with the star topology as the initial solution when the number of nodes is low, and then the "best link to root" method gives results that are similarly as good when the number of nodes increases. The link quality goes higher when the number of nodes increases on this plot, because the more nodes, the more the system has opportunities to select the better links. In the remainder of this section, we use the *star* topology to build the initial solution.

LP vs SD method

In order to know if the SD technique is viable, we compared it to the LP technique in terms of weighted link PDR, and execution time required to build the solution. The radio link PDR was chosen randomly in the $[0.7; 1]$ interval, while for each number of nodes and for each strategy, 2000 initial situations were executed. Once again, the detailed simulation setup is presented in [88].

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [21], [88], [98] the detailed performance evaluation results are demonstrated.

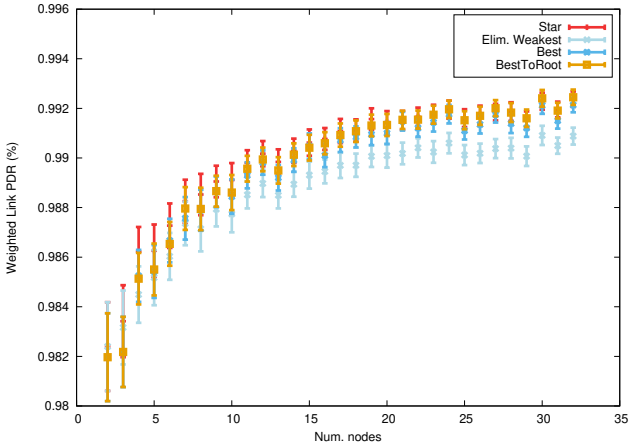
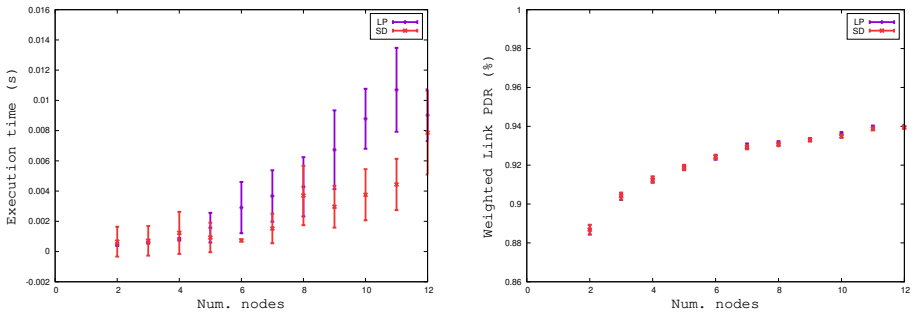


Fig. 4.10: Comparison of the weighted link PDR for the four initial solutions that have been tried, for a number of nodes from 2 to 32, with a 100 initial link situations in every case. *Taken from* [88].



(a) Link PDR in $[0.7; 1]$.

(b) Link PDR in $[0.7; 1]$.

Fig. 4.11: Comparison of topology building time (left), and average weighted link PDR (right) for LP and SD, for 2000 initial situations. *Taken from* [88].

In Fig. 4.11a, the execution time for both strategies is illustrated. As it can observed, the execution time becomes much longer for the LP method when the number of nodes increases. In Fig. 4.11b, the average weighted link PDR for both techniques is depicted. These results show that the LP technique comes with slightly better performance, however this difference does not seem to be significant. From all these results, we can conclude that even though both techniques are efficient, the the one based on SD is nearly as good as the one based on LP while it requires a lower computation time. Therefore, SD (with Star Topology as the initial solution) was the topology management technique we employed in our real-world experiments, see [98] (Chapter 5), with a network of 8 CC2650 I3Mote nodes in a Renault Fluence battery pack, see Fig. 4.12.

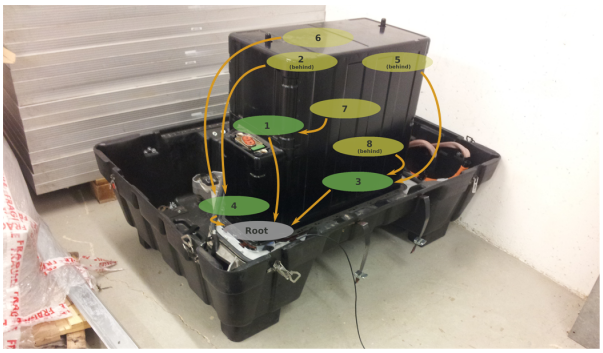


Fig. 4.12: Network topology at the end of the test with regard to the nodes' actual location. R_1 nodes are in green and R_2 nodes are in yellow. The arrows show the topology. *Taken from* [98].

4.4 Summary

The purpose of this Chapter was to provide a scheduling function that efficiently organises the transmission and reception timeslots in order to achieve high end-to-end reliability and bounded latency in a low-power wireless mesh network.

To bootstrap our research, we first conducted a thorough performance evaluation study on the standardised ***distributed*** scheduling function, i.e., ***MSF***. Our results showed that the convergence pattern of MSF has a large impact on the number of packet losses and usage of resources. On one hand, we observed that MSF allocates new resources at a rate which depends on the number of the already allocated cells. During this allocation process, packets exceeding the current transmission capacity are enqueued and dropped when the queue is full. On the other hand, MSF also leads to overprovisioning, i.e., more cells are allocated than what is actually required to carry the current traffic load. Indeed, moderate overprovisioning is beneficial as it allows to quickly absorb small, transient increases in traffic load without triggering new cell allocations. However, even with a constant rate traffic, the amount of overprovisioning in MSF is significant, thus wasting a large part of the available network resources.

Therefore, we then proposed the ***LDSF***, a ***distributed*** scheduling function. Instead of classical approaches, LDSF does not need to reschedule all the cells toward the destination when the quality of a specific link changes. LDSF is rather designed to handle the worst case scenario, provisioning enough cells for retransmissions. It divides the long slotframe into small blocks that repeat over time. Each node selects the right block corresponding to its hop distance from the border router to minimize the delay. Moreover, ghost cells are automatically reserved in the slotframe to cope with retransmissions. Besides, a device stays awake during these ghost cells only if the previous transmission has failed, to save energy. Our simulation results demonstrated that LDSF can achieve a low latency and jitter with high reliability, even for multi-hop topologies.

Finally, we proposed two ***centralised*** strategies for network management that enable very high network reliability even under real-world conditions such as the BMS in EV. The first method is based on ***LP***, and while it does generate a highly reliable network topology, it can be costly in terms of processing time in some scenarios. The second method, based on ***SD***, provides similar to LP results in terms of network reliability with a much lower computation time.

Chapter 5

Multi-path Strategies in RPL-based Wireless Mesh Networks

A reliable and available network ensures that data packets traverse the multi-hop network, in our case the RPL network, in a bounded window of time throughout the network lifetime and independently of the potential network congestions or external interferences.

As previously detailed, the IEEE Std 802.15.4-2015 TSCH protocol comes with resource reservation where the packet transmissions and receptions are scheduled. However, wireless links in low-power and constrained networks are lossy by nature, and moreover they are heavily affected by external interference and noise. Therefore, typically the wireless communication comes with retransmission schemes, but at a cost of energy consumption, latency, and bandwidth, since additional timeslots are required. In TSCH, if a frame transmission fails, then the transmitter in order to retransmit the very same frame will have to wait for few timeslots, or even for the whole slotframe, e.g., 101 timeslots according to the standard [20], which is more than a second. In the extreme case when a node crashes, or in the case of over-the-air-programming, the link quality between two devices will significantly decrease for some time, which will essentially increase the packet losses in the multi-hop network. In such a scenario, even retransmission schemes over different radio channels will not allow the packet to pass through this wireless link. At the network layer, RPL comes with a failure solution wherein a child node will select another parent. However, the time required for failure detection and new parent selection is large [67], and during this time, many data packets will be discarded. Unfortunately, these protocols, do not provide a high level of Quality of Service (QoS) since they depend on the variability of the link quality and the availability of the selected relay nodes.

Multi-path routing protocols have been a popular approach over the past years for different reasons, including to mitigate traffic load and enhance end-to-end network reliability. Indeed, they enable multiple paths using different nodes from a source to a destination, and depending on the use case, these multiple paths can be used alternatively or simultaneously. The conducted work is based on the RPL routing protocol, and extends it with multi-path redundancy.

In this Chapter, we first introduce the Packet Automatic Repeat reQuest, Replication and Elimination, and Overhearing (PAREO) functions that provide the necessary tools to support multi-path in RPL. We then present a series of novel contributions which take advantage of path diversity and data duplication to combat the potential losses and to achieve bounded latency in low-power wireless mesh networks. The contributions presented in this chapter demonstrate that reliable and available networking can be ensured by using multiple parallel paths instead of retransmissions over the default single path.

These works have been conducted during the ***PhD of Tomas Lagos Jenschke***, the ***Postdoctoral period of Remous-Aris Koutsiamanis*** as well as during the ***research internships of Ana Czarnitzki Estrin, Julian Martin Del Fiore, Maurine Kersale, and Tadanori Matsui*** whom I had the honour to supervise.

5.1 The PAREO Functions

The PAREO methodology consists of a set of functions with the aim of improving network reliability and availability in low-power wireless mesh networks [14]. The individual functions, illustrated in Fig. 5.1 and 5.2, complement each other towards this goal but work somewhat independently from each other.

The main objective of PAREO is to ensure high reliability and fault-tolerance in the presence of temporarily unavailable nodes while also minimizing latency and jitter. To achieve this goal, PAREO takes advantage of the physical properties of wireless technologies via packet replication and elimination as well as overhearing, thereby increasing the number of opportunities for a packet to reach its destination.

The result is a trade-off regarding energy consumption, which is increased to achieve the previous aims. The main reason for this is the overhearing operation, and specifically due to our implementation which set all nodes in the Parent Set (PS) of a node in the listening state. A more energy efficient but complicated solution would only set the Alternative Parent (AP) node in the listening state.

In the following subsections, I will present each constituent PAREO function and how they interact.

Automatic Repeat reQuest Function

The ARQ function performs the re-transmission of a data packet when a previous transmission failed [66]. In our context, we employ a link layer ARQ so the decision for re-transmission is local to the transmitting node. This function requires the use of an ACK control packet for each data transmission. In TSCH, each unicast transmission is by default configured to require an acknowledgement, thus the lack of its reception can be interpreted as a failed transmission.

Since in our context only one packet will be "in flight" at any one point in time in a given cell, we can use the Stop-and-Wait variation of ARQ, as described below. The function starts by sending a packet and setting a short timeout to await an ACK (within the same cell). If the ACK is received from the receiver, the transmission is marked as successful. If no ACK is received, then the transmission is marked as temporarily failed and is re-scheduled again in the future. For the same data packet, each time a Retransmission (RTX) fails,

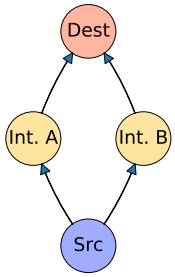


Fig. 5.1: TSCH example network topology. *Taken from* [14].

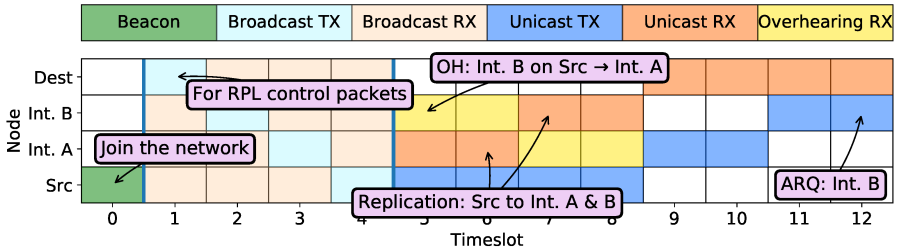


Fig. 5.2: TSCH example schedule showing Replication, Overhearing, and ARQ operations. *Taken from* [14].

a counter is incremented. If the counter reaches a threshold, named the RTX count, then the packet is marked as permanently failed and it is not scheduled again.

The scheduling of the re-transmission is tightly coupled with the structure of the TSCH schedule. Depending on the schedule, the timing of re-transmission attempts can vary significantly: from immediately after a temporary failure to later during the next repetition of the slotframe. While it is possible to consider almost "any" number of re-transmission attempts for each pair of nodes in one slotframe, the more the reserved cells for potential re-transmissions, the larger the slotframe size.

Replication and Elimination Functions

The Replication and Elimination (RE) function modifies packet forwarding to send a packet not only to the PP of a node but also to other nodes in the PS [45]. In the default implementations of RPL, each node only uses one node to forward packets (i.e., the PP). The problem with that approach is that if the PP fails for any reason, all packets from its children will be discarded until a new parent is selected. However, selecting a new parent requires one of two time-consuming processes, either local or global repair, and during this period connectivity is not available. If the queue fills up, and the connectivity not restored, the data packets will start being dropped. RE do not avoid the local or the global repair to select a new PP. However, having established two or more parents, the connectivity is maintained and, thus, packet drops are avoided due to an inaccessible parent node.

In order to make this possible, a data packet is cloned and each copy is forwarded over a different parent node, called an AP, in the PS. Performing packet replication creates multiple copies of the same data packet, which traverse the multi-hop network independently. In order to avoid a situation where each received copy results in a further independent set of packet copies being forwarded, resulting in flooding, a packet elimination function is employed to remove unnecessary duplication.

To eliminate duplicates, a way of identifying data packets is required, and usually some type of tag is used as part of the packet header.

The interaction of RE with the TSCH schedule only concerns the required existence of cell(s) for transmitting packets from a node not only to its PP but also to the AP.

Overhearing Function

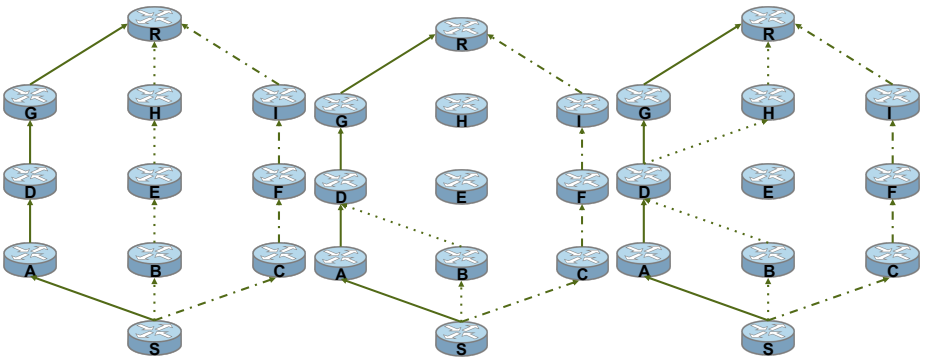
The Overhearing (OH) function [99] can be used to take advantage of the shared nature of the wireless medium in order to further increase network end-to-end reliability. This function allows a node to simultaneously forward a data packet to multiple parent nodes, which by virtue of being in the PS are assumed to be within radio range. The OH function works in addition to RE to increase reliability without sacrificing transmission latency.

Since with this function the sender broadcasts the packet and receiving nodes disable their MAC filter, two issues are required to be addressed:

- Firstly, promiscuous OH is performed for data packets which are typically forwarded with ACK-ed unicast transmissions. However, multiple parents may receive the broadcast and attempt to respond with an ACK, potentially resulting in a collision if multiple nodes attempt at the same time. This would lead the transmitting node to consider the transmission to have failed. Thus, it is important that only one node, i.e., the recipient indicated in the destination address field, responds with an ACK, while all other nodes silently receive the packet without sending an ACK. This capability is neither present by default in TSCH nor implemented in Contiki OS. We provided a solution by introducing OH cells, which accept packets sent to any destination (by disabling MAC filtering) but which suppress responding with an ACK even when one is requested.

- Secondly, it is desirable to control the replication of packets via OH so that only the AP node forwards the data packet, and not any node that happens to overhear it. This is not supported in TSCH or RPL by default, thus we have evaluated two options to implement it. One way to do this is via the TSCH schedule, configuring only the AP to overhear when the node transmits. Another is by storing the address of the AP in the packet so even if other nodes receive it, they can check the stored AP address against their own and avoid forwarding. In our implementation we have opted for the latter since it simplifies the TSCH schedule management.

The interaction of OH with the TSCH schedule only concerns the required existence of receiver cell(s) for OH packets from a node to its AP.



(a) Fully disjointed scenario. (b) Disjoint pattern with merging paths. (c) Disjoint pattern with nested N-disjoint.

Fig. 5.3: Potential N-disjoint scenarios. *Taken from* [15].

5.2 n-Disjoint Strategies

In this section, two different n-Disjoint algorithms are presented [15]. The first one is called "Default Strategy", where only the source node transmits n copies plus the original data packet to $n+1$ parents. The second algorithm is called "Advanced Strategy", where upon a potential path merging, the "merging point" node forwards each of received replicas to a different parent.

Default Strategy

In the "default strategy", only the source node transmits multiple replicas (i.e., replication function) of the same data packet in disjoint paths. The source node selects the $n+1$ best (in terms of ETX) parents given its PS, where n is the number of replicas. Then, it sends a copy of the data packet to these $n+1$ best parents. The relay nodes forward the received packets to their PPs, again based on the default parent selection process of RPL, up to the DODAG root. Note that this is valid for $n < |PS|$. Moreover, the ARQ function is used in all wireless links with a bounded number of RTX. The ideal case of this implementation, where all replicas follow completely disjoint paths and reach the final destination, is illustrated in Fig. 5.3a.

However, two or more disjoint paths may merge on one relay node, as that node may be selected as the PP by several nodes. As a result, it is probable that fewer paths (less than $n+1$) are actually employed. In Fig. 5.3b, such a scenario is depicted where even with perfect link qualities, the root node may not receive $n+1$ replicas, since some of the paths may merge. Nodes A and B have the same PP, node D, and, thus only one replica is forwarded from node D. The next algorithm addresses this issue to maintain the original number of paths in the network.

Advanced Strategy

To overcome the previously presented issue, an advanced algorithm was designed to detect when path merging takes place, and to allow the received copies to

follow different paths. Indeed, in case a relay node receives two or more copies of the same packet from different children, it will forward them to different parents. This technique handles path merging scenarios, however, it does not recreate replicas of data packets that were lost due to issues related with wireless link quality. In Fig. 5.3c, such a scenario is illustrated, where the node D forwards the two received copies from nodes A and B to nodes G and H.

In the Advanced Strategy, each of the relay nodes selects its $n + 1$ best parents given its PS, and sorts them in ETX order, where the first would be the PP. If a replica is received at the relay node, after the original copy, it forwards it to the next best parent that was previously stored. As in the Default Strategy, this is valid for $n < |PS|$. Moreover, it should be noted that, as with the Default Strategy, the ARQ function is employed at each wireless link.

5.3 Common Ancestor (CA) Algorithms

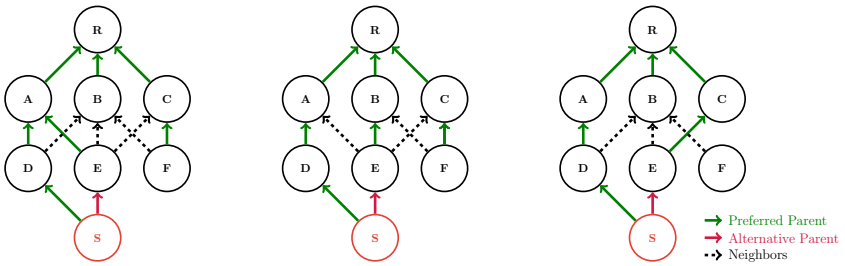
While there are many ways of implementing this advanced strategy of RE on in-between nodes in addition to the source node, we have identified a family of algorithms which produce a braided routing pattern with desirable properties. In this section, parent selection algorithms based on a braided pattern [100] with two routing paths are presented [16]. In the main path, nodes forward packets via the PP, while in the alternative path nodes employ another node called the AP. The AP selection algorithm greatly impacts the number of nodes that will be part of the forwarding process, and thus the energy consumption, i.e., the more the nodes involved, the higher the energy consumption. The Common Ancestor (CA) approach selects an AP if there is at least one potential parent in common between the parent sets of the PP and the AP. Indeed, the goal behind this strategy is to select an AP that will avoid (or reduce) the potential flooding, and concentrate the forwarding efforts towards a single direction. Given this trade-off between reliability and power consumption, three algorithms were defined, which vary in their flexibility when choosing an AP.

Strict CA

Based on the Strict CA approach, node k selects a candidate parent node c as an AP if the candidate node's PP is the same as the PP of the PP (the preferred grandparent) of the current node k, i.e., $PP(PP(k)) = PP(c)$. The Strict CA algorithm is illustrated in Fig. 5.4a. Considering that nodes D and E have the same PP (node A), then E is a candidate node to be selected as AP for the current node S. Therefore, since node E is in the PS of S, E is selected as AP.

Medium CA

Node k by employing the Medium CA algorithm selects a candidate parent node c as an AP if the candidate node's PS contains the PP of the PP (the preferred grandparent) of the current node k, i.e., $PP(PP(k)) \in PS(c)$. The Medium CA algorithm is illustrated in Fig. 5.4b. As it can be observed, none of the nodes have a common PP except for those connecting to the root. However, since E has A, B, and C in its PS and D has A as PP, E can be selected as the AP of the node S.



(a) Strict CA. (b) Medium CA. (c) Soft CA.

Fig. 5.4: Common Ancestor Algorithms to select an AP. *Taken from* [16].

Soft CA

Finally, node k by employing the Soft CA algorithm selects a candidate parent node c as an AP if the candidate node's PS has any common node with the PS of the PP (the preferred grandparent set) of the current node k, i.e., $PS(PP(k)) \cap PS(c) \neq \varnothing$. The Soft CA algorithm is illustrated in Fig. 5.4c, where E is selected as AP of node S since its PP D has B as a common PS node with E.

In cases where multiple nodes are possible candidates to be selected as AP, then for all CA algorithms, the node with the best parent selection metric will be selected, e.g., lowest rank when RPL is employed.

Common Ancestor Trade-offs

As it can be seen in Fig. 5.4, in each algorithm, the PS of the PP of a node directly affects which nodes are eligible as APs. Even with the Soft CA algorithm, at least one node from the PS of the candidate AP has to belong to the PS of the PP of the node.

Considering a regular grid-like topology of $L \times N$, as shown in Fig. 5.5, where L is the number of layers between the source node (node S) and the DODAG root (node R), and N is the number of nodes per layer, the probability of obtaining an AP according to each algorithm is given in Fig. 5.6, parametrised over the PS set size N and the PS_{MC} set size M. Indeed, the nodes report a subset of their parent set in the PS extension of the Node State and Attribute (NSA) object in the Metric Container (MC) of DIO messages, i.e., PS_{MC} [39]. Our proposed PS_{MC} extension, currently in the process of being standardised at the IETF [39], contains a fixed number of IPv6 addresses M, with $1 \leq M \leq N$, where N is the number of nodes per layer. The probability of being able to find an AP depends on the employed AP selection algorithm and the values of M and N.

As expected, the Strict CA algorithm has a lower probability of finding an AP when comparing against the Medium and Soft algorithms since the probability of obtaining an AP is subject to sharing a common PP. Consequently, this performance is directly related to the network end-to-end reliability, since fewer APs can be selected and, thus, less redundancy than the other two CA algorithms is achieved in the network, however less energy is consumed as well. On the other hand, the Soft algorithm might achieve higher end-to-end network

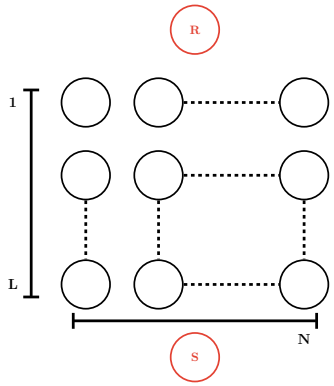
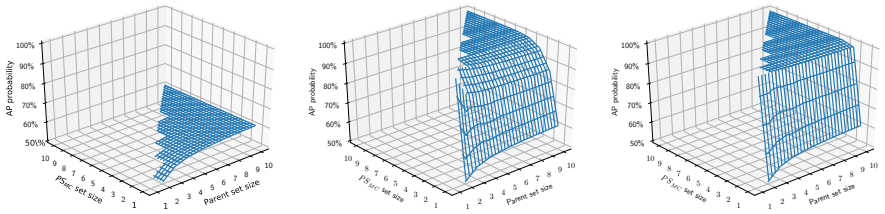


Fig. 5.5: Grid-like network topology, where node S is the source node, and R is the DODAG Root. The L intermediate layers each contain N nodes. Each node in each layer i is connected to all the nodes in the layer preceding it $(i-1)$. *Taken from* [16].



(a) Strict CA probability (b) Mediun CA probability (c) Soft CA probability

Fig. 5.6: Probability of finding an AP through CAs parametrised over the PS set size N and the PS_{MC} set size M. As it can be observed, as the size of PS of a node increases, the probability of obtaining an AP increases. *Taken from* [16].

reliability since its AP selection works by matching any node between the PS of the PP with the PS of a potential AP. Therefore, the number of nodes employed in the forwarding process might increase significantly (it might even lead to flooding), which however presents higher energy consumption.

It should be noted that contrary to Medium and Soft, the Strict method does not use the information from the set of parents, i.e., the M size of the PS_{MC} does not affect the probability, because it only requires the PP information that is propagated in the DIO message.

5.4 ODeSe: On-Demand Selection for Multi-path RPL Networks

As previously stated, for all three CA algorithms, the PS of the PP of a node k directly affects which nodes are eligible as APs. Even with Soft CA, the least restrictive selection algorithm, at least one node from the PS of the potential AP has to belong to the PS of the PP of the node k.

Considering the Strict CA, the algorithm that has the lowest probability of obtaining an AP given its selection policy, the number of parallel routing paths

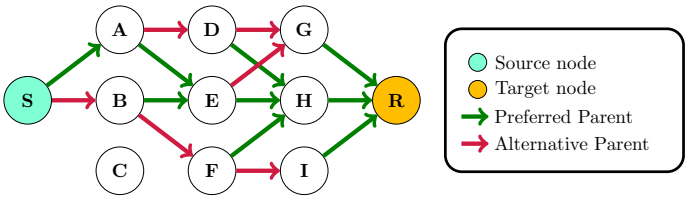


Fig. 5.7: Flooding using the strict CA algorithm. *Taken from* [101].

in the network is lower when compared against the other two CA algorithms. However, even under ideal conditions, when employing Strict CA the wireless mesh network can be flooded just as with the Soft CA algorithm.

Indeed, as it is depicted in section Fig. 5.7, where the topology is inspired from 5.3, where $L = 3$ and $N = 3$, the resulting routing paths include almost all nodes with the exception of node C. Furthermore, as it can be observed, the routing paths are spread out between the transition from relay nodes {A, B, C} towards {D, E, F}. This is mainly because even though each PP and AP has the same PP, this policy is not extended for APs. More specifically, nodes A and B have node E as PP, however, nodes D and F are selected as AP since they have node H as PP. This example demonstrates that even though Strict CA is the algorithm with the minimum routing spread among the CA algorithms, it may still generate a similar traffic load to its peers.

ODeSe Algorithm

While the Strict, Medium, and Soft CA algorithms perform well, the highest transmission reliability is achieved with Soft CA but at the cost of higher energy consumption due to higher flooding. In this section, we present On-Demand Selection (ODeSe) that aims to maintain the same level of end-to-end network reliability while controlling the flooding issue. The main difference between the CA algorithms and ODeSe is that with ODeSe each node decides not only its own (PP and AP) but also the PP and AP of the next hop. In other words, each node has a two-hop forwarding control.

By employing the ODeSe algorithm, at every hop, each node k selects its PP and AP using the Strict CA algorithm. In the case no valid AP node is found, then the node k will opt for a Medium CA strategy and a Soft CA if the Medium fails as well. Then, in the to-be-forwarded data packets, i.e., the replicated data packets, *each* node k includes the IPv6 addresses of the PP and the AP of its PP. More specifically, these two addresses are carried in the Hop-by-Hop option header of the IPv6 packet, see Fig. 5.8.

Once received at the next hop node(s) k', i.e., $k' \in \{PP(k), AP(k)\}$, the PP and AP of k, the following steps are taken. The k' checks if the carried PP address HbH_{PP} is indeed a valid PP for it, i.e., if the rank of the HbH_{PP} is lower than the rank of k', and if yes, it employs HbH_{PP} as the new PP for this specific data packet, see node E in Fig. 5.9b. Then, the original PP is restored after completing the forwarding operation for this data packet. The validity check is required because information at node k might be outdated due to not receiving up-to-date parent set information from k' via a DIO control packet. On the other hand, if the HbH_{PP} is not a valid PP, then the default PP of node k' is used, see Fig. 5.9c.

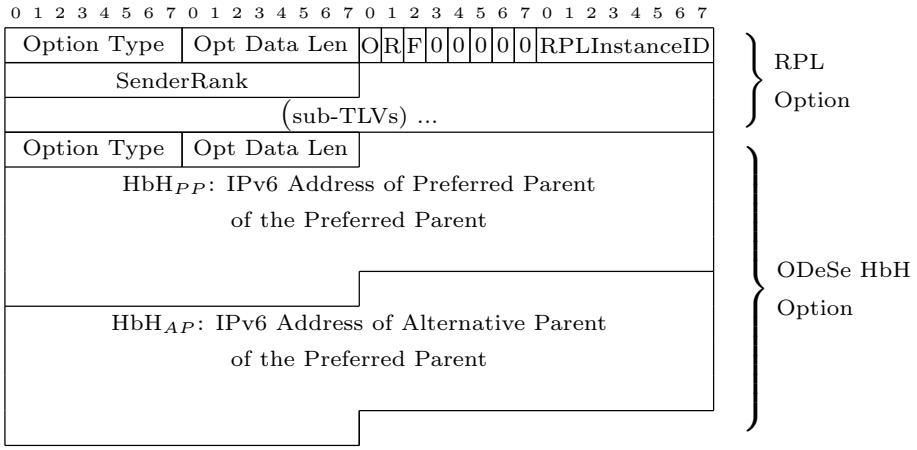
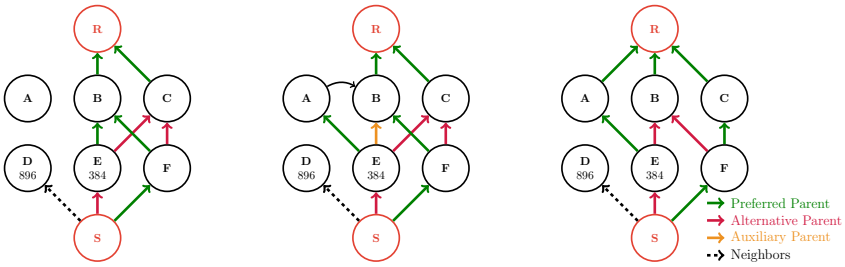


Fig. 5.8: IPv6 Hop-by-Hop option fields (RPL and ODeSe option). *Taken from* [13].



(a) Optimal scenario. (b) Parent change scenario. (c) Route adaptation.

Fig. 5.9: ODeSe functions for parent selection. *Taken from* [13].

Regarding the AP, node k' checks if the carried AP address HbH$_{AP}$ is also a valid AP by employing the Strict CA policy. If it is, then k' uses HbH$_{AP}$ as the AP, see Fig. 5.9a. If it is not, then k' will try to obtain a new AP using first Strict CA, and if that fails, it will fall back to the Medium CA and then Soft CA strategy.

When k' has selected the AP and PP for the next hop, it does what node k at the previous hop did before it. It stores and replaces the addresses of the PP and the AP of its PP, i.e., $\text{HbH}_{PP} = \text{PP}(\text{PP}(k')), \text{HbH}_{AP} = \text{AP}(\text{PP}(k'))$ in the to-be-forwarded data packets. Thus, the entire process repeats until the data packet is delivered to the target node.

5.5 Performance Evaluation

Simulation Setup

We employed the Contiki 3.0[1] operating system [102] and the COOJA simulator to implement the PAREO functions, and the n-Disjoint, CA and ODeSe

[1] https://github.com/ariskou/contiki/tree/draft-ietf-roll-nsa-extension

algorithms with a directed graph radio medium. Contiki comes with the default RPL routing protocol implemented, which is the one used for evaluating the single-path with ARQ approach.

The network topology used for the evaluation is the one depicted in Fig. 5.5 where $L = 5$ and $N = 6$. The radio links in this topology come with 50% link quality on average, and every node (except the root and its direct children) has 6 potential parents to forward data. This link quality value was chosen following the Dust Networks[2] definition of a healthy network where each device should have at least a 50% link quality with its neighbors and at least 3 potential parents.

At the MAC layer, TSCH with a timeslot length of $10ms$ and a slotframe size of 357 timeslots, i.e., $3570ms$, was employed in order to provide a sufficient number of timeslots, i.e., to allocate 1 RTX for each original data packet transmission. Furthermore, a centralized TSCH scheduler was employed [103], similar to one presented in 5.1, see Fig. 5.2.

Regarding the traffic in the network, in each simulation 250 UDP packets were transmitted from the source node S to the root node R, with no fragmentation. The packets were sent once every 15 seconds so that at any one point in time there is only one packet being forwarded in the network.

Finally, at the routing layer, we evaluated all proposed algorithms, n-Disjoint, CA and ODeSe, against the default RPL single-path approach with different RTX configurations.

More specifically, the routing algorithms evaluated are:

1. Single-path (SP) with:
   (a) 0 RTX, i.e., no retransmission when an ACK is not received.
   (b) maximum 1 RTX when an ACK is not received.
   (c) maximum 3 RTX when an ACK is not received.
   (d) maximum 7 RTX when an ACK is not received.
2. n-Disjoint with the option of three parents i.e., 3P:
   (a) 0 RTX, i.e., no retransmission when an ACK is not received.
   (b) maximum 1 RTX when an ACK is not received.
   (c) maximum 3 RTX when an ACK is not received.
   (d) maximum 7 RTX when an ACK is not received.
3. Strict CA with 1 RTX maximum.
4. Medium CA with 1 RTX maximum.
5. Soft CA with 1 RTX maximum.
6. ODeSe with 1 RTX maximum.

For each of the above routing algorithms, we executed 20 simulations with different pseudo-random number generator seeds, to increase the statistical reliability of our results. In total, 5000 data packets were sent for each routing algorithm. The detailed simulation setup is presented in [101].

[2]SmartMesh IP Application Notes, Linear Technology Corp. 2012-2016.

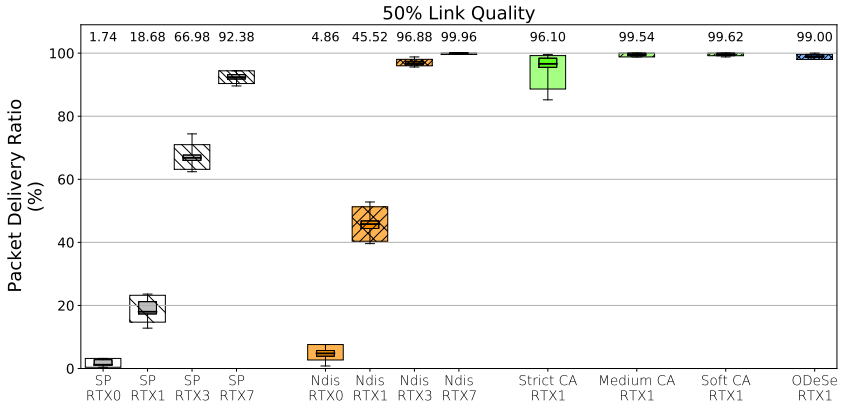


Fig. 5.10: End-to-end PDR performance. *Taken from* [13].

Simulation Results

In the following, the key take-aways are presented to demonstrate the overall tendency of the proposed algorithms. In [18], [17], [69], [16], [14], [13] the detailed performance evaluation results are presented.

PDR Performance

Fig. 5.10 shows the performance of all algorithms in terms of PDR. As expected, the performance of multi-path algorithms is essentially better than single-path. Then, as observed, the ODeSe and the CA algorithms come with higher PDR results by demonstrating the advantage of using the OH function, where more chances for a data packet to arrive at the DODAG Root are added. Indeed, the ODeSe and CA algorithms can reach these values mainly because the upstream data packet forwarding process involves a maximum of 8 transmission attempts towards a single direction. The PAREO functions allow for a packet transmission, a replication, and overhearing which enables both the PP and the AP to receive the same data packet. Considering that these actions are performed by each parent, a single node delivers a total of 4 transmission attempts to the PP and AP, therefore there are in total a maximum of 8 transmission attempts towards the DODAG Root.

Delay Performance

The slotframe size of the schedule severely impacts delay performance [14]. It is interesting to note that the main advantage of using PAREO, i.e., CA and ODeSe algorithms, is that it can achieve low delay and low jitter at the same time, as shown in Fig. 5.11. Both the CA and ODeSe algorithms provide approximately the same performance given the same schedule, but for single-path and n-Disjoint, once the number of MAC layer re-transmissions attempted surpasses the number of transmission opportunities in the same slotframe (i.e., single-path and n-Disjoint with RTX 3, and 7) both delay and especially jitter are significantly impacted. While it is possible to just increase the number of transmission opportunities in the schedule to reduce jitter, this would nevertheless impact delay.

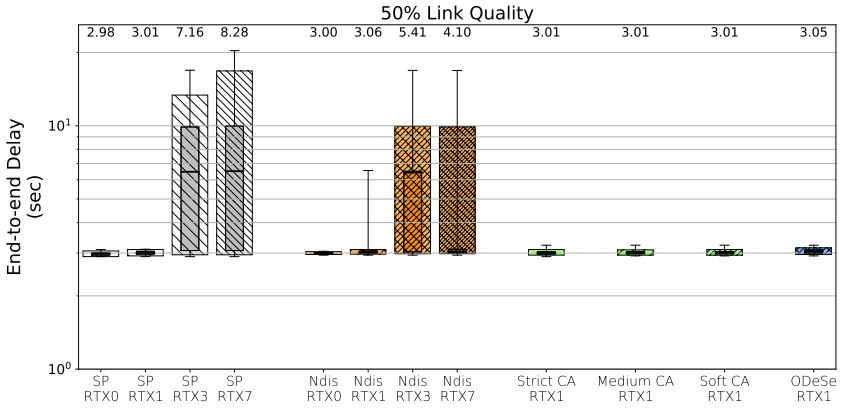


Fig. 5.11: End-to-end delay performance. *Taken from* [13].

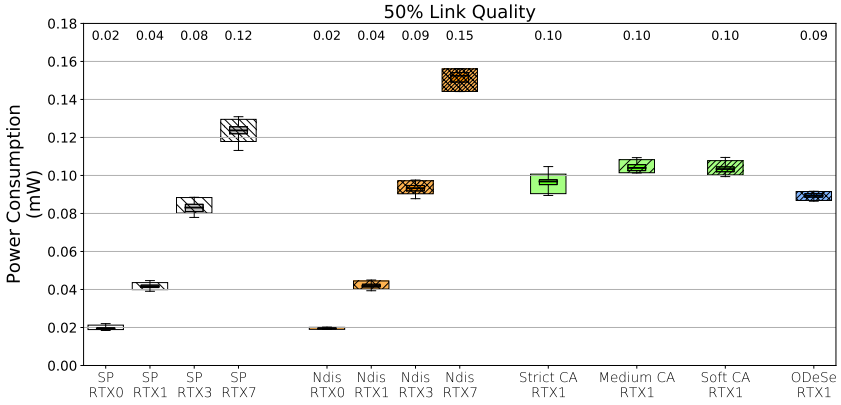


Fig. 5.12: Average power consumption per node (mW). *Taken from* [13].

Energy Consumption:

In Fig. 5.12, we report the energy consumption per slotframe, to allow meaningful comparisons. As it can be observed, depending on the RTX values, the single-path and n-Disjoint scenarios generally have comparable (or better) energy consumption with the CA and ODeSe, with higher RTX values predictably leading to higher energy consumption. The CA and ODeSe algorithms in general do not rely as much on retransmissions, and therefore have less room for energy consumption improvement due to this factor. This is mainly due to the number of nodes used to forward a single data packet and due to the overhearing operation that requires an extra timeslot. Among the multi-path algorithms, Soft CA comes with the highest energy consumption due it's production of flooding, while ODeSe consumes the least energy, since it concentrates its routes without dispersing them over the DODAG, see Fig. 5.13.

It should be noted that the energy consumption measured comprises the cost of sending both data and control packets.

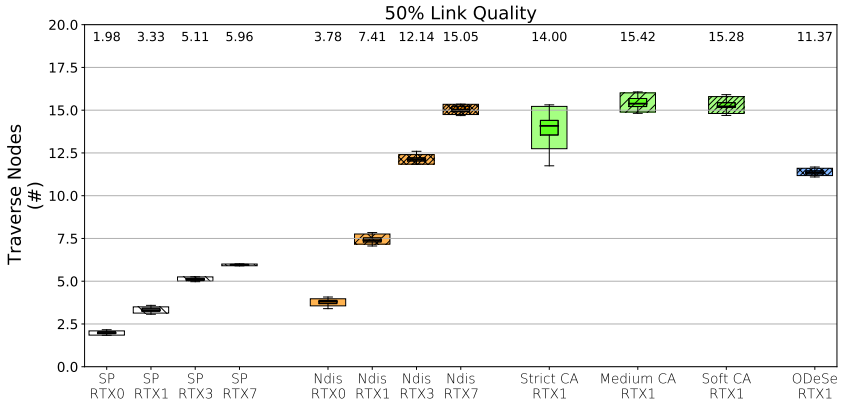


Fig. 5.13: Average number of relay nodes per packet. *Taken from* [13].

5.6 Summary

The main objective of this Chapter was to ensure high reliability and fault-tolerance in the presence of temporarily unavailable nodes or sudden drops of link quality while also minimizing latency and jitter in the **routing layer**. To achieve this goal, first the PAREO functions were presented which take advantage of the physical properties of wireless technologies via packet replication and elimination as well as overhearing, thereby increasing the number of opportunities for a packet to reach its destination.

Then, three multi-path strategies were proposed to achieve high end-to-end network reliability and low jitter performance.

First, we introduced two n-Disjoint multi-path algorithms, i.e., the Default and Advanced Strategies, which increase the network end-to-end reliability by taking advantage of the disjoint pattern, in conjunction with the RE and ARQ functions. Overall, it is possible to achieve remarkable results with the n-Disjoint algorithms. These results depend on the number of retransmissions and the number of replicas per data packet.

Then, the CA algorithm with three variations (Soft, Medium and Strict CA) was presented which achieves high network reliability using multi-path techniques based on a braided pattern but at the cost of high energy consumption due to flooding in the network. This led us to the ODeSe algorithm which achieves very high reliability with lower energy consumption, thus providing better options in the reliability - energy consumption trade-off. More specifically, the results show that it is possible to attain the high reliability offered by the highest energy consuming CA algorithm (Soft CA) with energy consumption lower than the least energy consuming CA algorithm (Strict CA). Additionally, ODeSe offers a better trade-off than the highest reliability single-path algorithm evaluated (Single-path RPL with 7 RTX). Therefore, ODeSe offers a novel solution for high-reliability industrial wireless networks.

Chapter 6

Conclusions & Perspectives

6.1 Overview

This manuscript has presented my main research axes which I have been pursuing since my PhD. I have had the chance to work with some of the smartest researches and engineers in the field of low-power wireless mesh networks (and not only), and through contributions presented in this manuscript (both in academic and standardisation communities), I hope to have had impact in providing reliable and available networking properties in critical application, including industrial applications. I hope you have enjoyed reading it, and to have been able to give a comprehensive overview. The research axes have been structured and presented in three themes, where each represents one layer from the LLN protocol stack.

First, we focused on the MAC layer. Indeed, we first conducted a series of experiments to characterise the IEEE Std 802.15.4-2015 radio channels in an indoor testbed. We demonstrated in particular the existence of specific per-link characteristics, where external interference may be locally high for some radio channels. Therefore, we worked on improving the efficiency of the slow channel hopping technique with blacklisting techniques, where the objective was to exclude the low-quality channels from the channel hopping sequence. Towards this aim, we first proposed a distributed blacklisting technique, that adopts a pseudo-random approach to avoid using the worst (in terms of link quality) radio channels. While this approach allows each radio link to select in a distributed manner the best radio channels to employ, collisions may still arise pseudo-randomly. Therefore, we then focused a centralised blacklisting scheme, able to adapt the blacklists for each radio link, while still making the full behaviour deterministic, by re-arranging the conflicting blacklists. Moreover, we proposed an hybrid blacklisting scheme that exploits the full radio spectrum, assigning multiple channel offsets per timeslot to increase the network efficiency when handling long blacklists.

Next, we tackled with the scheduling function layer. We first performed a thorough performance evaluation study on the behaviour and reactivity of the MSF, the standardised scheduling function. Our results demonstrated that the convergence pattern of MSF is the root cause of the majority of packet losses observed in the network. Moreover, we showed that MSF is prone to over-

provisioning of the network resources, especially in the case of varying traffic load. Therefore, we proposed the LDSF, a distributed scheduling function, that aims to meet the requirements for reliable and low latency networking even when radio links are subjected to external interferences. The proposed solution relies on the organisation of the slotframe in smaller parts, called blocks, where each transmitter selects the right set of blocks, depending on its hop distance from the root, so that retransmission opportunities are automatically scheduled. We also proposed two algorithms for centralised network management when considering the BMS application, based on the linear programming and simple descent techniques, in order to optimise the topology and resource allocation. We have tested this centralised approach with a network of 8 CC2650 I3Mote nodes in a Renault Fluence battery pack.

The last Chapter is dedicated on the routing layer. The conducted work was based on the RPL, the de facto routing protocol for LLNs, and we extended it with multi-path redundancy. We first introduced the PAREO functions that provide the necessary tools such as packet replication and elimination as well as overhearing to increase the number of opportunities for a data packet to reach its destination. We then presented a series of novel contributions, i.e., n-Disjoint, CA, and ODeSe, which take advantage of path diversity and packet replication to combat the potential losses and to achieve bounded latency in low-power wireless mesh networks. The contributions presented in this chapter demonstrate that reliable and available networking can be ensured by using multiple parallel paths instead of retransmissions over the default single path.

6.2 Perspectives

The more low-power wireless mesh technology evolves, the more potential it creates both in the industrial sector and in academic research. In the following sections, I highlight the high-risk-high-gain research axes, which I plan on passionately investigating. In light of this work, there are upcoming new challenges for me, constituting some natural evolutions.

SDN Approach in Low-Power Networks

Even though my colleagues and I have done many contributions over the years on achieving reliable and available properties in low-power wireless mesh networking, however, we are still far from the envisioned architecture. So far, I have work on three research axes in parallel, or in semi-chronological order, *i*) radio channel blacklisting, *ii*) scheduling functions and *iii*) redundant multi-path routing. However, I did not have the opportunity to combine all these three axes over the same low-power wireless mesh network. Indeed, I have only contributed by combining a centralised schedule (e.g., PCE) that builds a static schedule based on TSCH [103] in conjunction with a distributing multi-path routing algorithm based on RPL [13], [16], [17]. Under such cross-layer architectures, there are unexpected issues to deal with, for instance when the dynamic routing protocol auto-adapts to the link quality changes (e.g., when the nodes in the network changes its parents), while having a schedule that is based on a centralised entity, it might take time to converge again the resource allocation.

Furthermore, in an effort to add programmability and virtualisation, and while observing the continuous increase of interest on Software-Defined Networking (SDN) in traditional networks and lately in IoT [104], I am convinced that in order to achieve the envisioned architecture that manages and combines efficiently the proposed functions and algorithms of this manuscript, then the proposals must be redesigned, redeveloped and redeployed based on a centralised entity. SDN is a centralised network where the intelligence is taken away (or reduced) from the intermediate and end nodes, and it is managed by a controller that has a global view of the network which distributes to the nodes the necessary rules to handle the communication and networking operations. A rule is an instruction that indicates the node how to handle a received packet. It contains a matching criteria detailing to which packets the rule applies to, and an action to perform on the packet.

The main possible actions are forwarding a packet to a next hop, dropping the packet or modifying its fields. Thanks to these rules, the software running on the nodes can be significantly less complex.

The IoT community has taken interest in this SDN paradigm, and several implementations have been released for low-power wireless mesh networks with goals of improving the QoS [105] and the flexibility of these networks [106], [107]. However, the addition of this controller introduces new challenges for these constrained wireless mesh networks. Indeed, the lossy nature of the radio links decreases the end-to-end reliability between the nodes and the controller, the limited memory of the intermediate and end nodes makes them unable to store a large number of rules, and additionally the control plane overhead is costly in energy and traffic. Still, the recent studies provide hope that an optimised SDN would outweigh these drawbacks by improving the network reliability, providing availability and reducing the energy consumption.

The current SDN propositions for low-power wireless mesh networks only focus on making the network layer programmable, mimicking the traditional networks. However, I am envisioning to design an architecture where the whole stack is programmable, including the resource allocation sublayer (i.e., SF), MAC layer (i.e., TSCH with radio blacklisting capabilities), and the PHY layer, thanks to the the emerging Software-Defined Radio (SDR) technology [108],[109] in conjunction with the multi-antenna constrained devices [110] [8]. Indeed, the goal is to investigate the possibility of designing and developing SDN nodes close to finite-state machines whose sequences of events are decided by the controller of the network.

Another important focus of my work is the respect of the separation between the routing time scale and the forwarding time scale described by the IETF RAW working group. The radio links in low-power wireless mesh networks can have short term link quality variations (i.e., flapping) because they use unlicensed frequencies where interference can occur. When a node has to transmit a packet but the link allocated by the controller is not available, the node has no way to ask the controller for another route and still transmit the packet in due time, it is too late. Therefore the nodes must have a degree of adaptability. Adding this adaptability while keeping the SDN philosophy of rules requires the node to be stateful and to keep short term information about the network characteristics. I am studying the possibility to make this a reality and working on how the nodes gather and store information as well as how controllers give them conditional rules to adapt to varying network conditions.

For this research axis, I am working with two PhD students, who both started in 2020.

Multi-hop & Multi-path Routing Strategies in Integrated Access and Backhaul for 5G

Towards wireless backhaul, 3rd Generation Partnership Project (3GPP) standardised Integrated Access and Backhaul (IAB) in Release 16, where the goal is to reuse the existing framework of Fifth Generation Mobile Networks (5G) radio air interface for backhaul purposes by intelligently multiplexing access and backhaul in the time, frequency and/or space domain [111]. By employing such an approach, the potential operators may improve the cellular coverage by installing denser networks with IAB nodes, without having to lay fiber for backhaul. This technology has generated a lot of interest in the industry since IAB reduces the costs of very dense deployments and improves cellular coverage [112].

The mmWave spectrum is a promising enabler for small cell deployments (i.e., IAB) where there is no existing wireline infrastructure. However, at such high frequencies, there is high propagation loss which limits the communication range of mmWave base stations [113], and, moreover considering the blockage phenomenon due to presence of obstacles which reduces essentially the received signal power [114], advocates for a high-density and multi-hop deployment of IAB base stations using mmWave frequencies [115], [116].

In the novel multi-hop multi-path IAB networks, the existing solution-functions, i.e., Hybrid ARQ (HARQ) at the PHY layer and ARQ at the Radio Link Control (RLC) layer, for end-to-end reliability and bounded latency are challenged. While these techniques are excellent for guaranteeing single-link reliability, they are not designed with a consideration for multi-hop multi-path network topologies since they cannot make use of multi-path diversity.

The IAB (multi-hop) network assumes a Directed Acyclic Graph (DAG) topology, similar to RPL, in which multiple paths can exist between the IAB donor and an IAB node. In case of backhaul link failure or IAB node congestion, this path redundancy feature can be applied for back-up purposes. Furthermore, it is also possible that the redundant paths between the source and destination IAB nodes are employed concurrently, to enhance end-to-end reliability or support load balancing [117]. The path establishment and selection should consider the long-term network performance, however more dynamic routing and forwarding decisions should also be made possible to support transmission of latency-sensitive traffic across backhaul links during potential short-term link flapping and blocking.

All these requirements and envisioned architecture of IAB match well with the contributions that I have been working both in academia and in standardisation bodies. Indeed, the PAREO functions 5.1 have been proved [14], [13] to be compatible solution at the routing layer, and very good candidate to be applied to the IAB architecture. Moreover, as a second step, it would be very interesting to extend the PAREO functions by incorporating more solutions that I have not yet found the time to investigate such as network coding and extended FEC algorithms.

Security Challenges in Low-Power Networks

IoT and low-power wireless mesh networks are increasingly deployed in the real world, but their security lags behind the state-of-the-art of non-IoT systems. As you have well observed, I have not mentioned much about the security considerations in low-power wireless mesh networks in this manuscript. Even though I have been involved in security-related research projects, I still do not consider myself an expert in this domain. Therefore, one of my research avenues is to further work and explore this field.

I have successfully supervised a relevant PhD thesis [118]. The main goal of the PhD was to improve the resilience of constrained IoT network components through the use of the Moving Target Defense (MTD) paradigm. MTD is a cyberdefense technique that perpetually randomises system components, with the intention of thwarting cyber attackers that previously relied on the static nature of them. Even though MTD has been implemented in conventional networks, its popularity in low-power wireless mesh networks is still lacking in the literature [119]. Throughout this PhD thesis, we have *i*) validated MTD as a suitable technique for IoT systems, *ii*) defined a modular distributed MTD framework that allows the instantiation of MTD techniques suitable for the constrained devices, and *iii*) proposed three concrete MTD mechanisms, two at the upper network layers (dealing with port-hopping and application RESTful interfaces) [120], while the third one at the physical layer employing Direct Sequence Spread Spectrum (DSSS) anti-jamming techniques [121].

These works were my first steps in exploring the field of security when it's applied in low-power wireless constrained networks. As it was identified in [119], network-based MTD strategies are predominant in the literature, however, there are still many opportunities that we identified to further explore. Indeed, the SDR and SDN technologies are particularly promising, once they are technically (and of course economically) possible for the wireless constrained networks, they may enable the implementation of MTD mechanisms at the physical layer [109], [108] and the logical topology level, respectively.

Furthermore, routing protocols for low-power wireless mesh networks such as RPL might also have robustness gains by randomising the potential moving parameters. For instance, randomising the routing metrics or even their values based on which node is selected as a "parent" toward the border router.

Bibliography

[1] C. A. Medina, M. R. Pérez, and L. C. Trujillo, "IoT Paradigm into the Smart City Vision: A Survey," in *Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2017, pp. 695–704.

[2] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019.

[3] Y. Kabalci, E. Kabalci, S. Padmanaban, J. B. Holm-Nielsen, and F. Blaabjerg, "Internet of Things Applications as Energy Internet in Smart Grids and Smart Environments," *MDPI Electronics*, vol. 8, no. 9, 2019.

[4] S. S. Reka and T. Dragicevic, "Future effectual role of energy delivery: A comprehensive review of Internet of Things and smart grid," *Elsevier Renewable and Sustainable Energy Reviews*, vol. 91, pp. 90–108, 2018.

[5] M. M. Dhanvijay and S. C. Patil, "Internet of Things: A survey of enabling technologies in healthcare and its applications," *Elsevier Computer Networks*, vol. 153, pp. 113–131, 2019.

[6] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[7] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[8] T. Chang, P. Tuset-Peiro, J. Munoz, X. Vilajosana, and T. Watteyne, "OpenWSN & OpenMote: Demo'ing A Complete Ecosystem for the Industrial Internet of Things," in *Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON)*, 2016.

[9] Cisco, "The Cisco Connected Factory: Powering a Renaissance in Manufacturing (white paper)," https://www.cisco.com/c/dam/r/en/uk/internet-of-everything-ioe/iac/assets/pdfs/cisco_ind_smart_rev2-wp.pdf, accessed: April 2021.

[10] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks," IETF, RFC 5673, October 2009.

[11] D. Chen, M. Nixon, and A. Mok, *WirelessHART: Real-Time Mesh Network for Industrial Automation*, 1st ed. Springer, 2010.

[12] ISA-100.11a-2011:, "Wireless systems for industrial automation:process control and related applications," *International Society of Automation (ISA) Std.*, vol. 1, May 2011.

[13] T. L. Jenschke, R.-A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, "ODeSe: On-Demand Selection for Multi-path RPL Networks," *Elsevier Ad Hoc Networks*, vol. 114, p. 102431, 2021.

[14] R.-A. Koutsiamanis, G. Z. Papadopoulos, T. L. Jenschke, P. Thubert, and N. Montavont, "Meet the PAREO Functions: Towards Reliable and Available Wireless Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[15] A. C. Estrin, T. L. Jenschke, G. Z. Papadopoulos, J. I. Alvarez-Hamelin, and N. Montavont, "Thorough Investigation of Multi-path Techniques in RPL based Wireless Networks," in *Proceedings of the 25th IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[16] T. L. Jenschke, G. Z. Papadopoulos, R.-A. Koutsiamanis, and N. Montavont, "Alternative Parent Selection for Multi-Path RPL Networks," in *Proceedings of the 5th IEEE World Forum on Internet of Things (WF-IoT)*, 2019, pp. 533–538.

[17] R. A. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, and N. Montavont, "From Best-Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4468–4480, 2018.

[18] G. Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, and N. Montavont, "Leapfrog Collaboration: Toward Determinism and Predictability in Industrial-IoT applications," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[19] V. Kotsiou, "Reliable Communications for the Industrial Internet of Things," Ph.D. dissertation, University of Strasbourg, 2020.

[20] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.

[21] G. L. Gall, N. Montavont, and G. Z. Papadopoulos, "Enabling IEEE 802.15.4-2015 TSCH based Wireless Network for Electric Vehicle Battery Management," in *Proceedings of the 25th IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–6.

[22] "Charter IETF CoRE WG, Constrained RESTful Environments," March 2021.

[23] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF, RFC 7252, June 2014.

[24] "Charter IETF ROLL WG, Routing Over Low power and Lossy networks," November 2019.

[25] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and A. R., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, RFC 6550, March 2012.

[26] "Charter IETF 6LoWPAN WG, IPv6 over Low power WPAN," March 2005.

[27] G. Montenegro, N. Kushalnagar, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," IETF, RFC 4944, September 2007.

[28] "Charter IETF 6TiSCH WG, IPv6 over the TSCH mode of IEEE 802.15.4e," March 2020.

[29] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LDSF: Low-latency Distributed Scheduling Function for Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, pp. 8688–8699, 2020.

[30] P. Thubert, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4," IETF 6TiSCH WG, Internet-Draft draft-ietf-6tisch-architecture-30 [work-in-progress], November 2020.

[31] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," IETF, RFC 8480, November 2018.

[32] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," IETF, RFC 8180, May 2017.

[33] T. Chang, M. Vucinic, X. Vilajosana, S. Duquennoy, and D. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)," IETF, RFC 9033, May 2021.

[34] D. Hauweele, R.-A. Koutsiamanis, B. Quoitin, and G. Z. Papadopoulos, "Pushing 6TiSCH Minimal Scheduling Function (MSF) to the Limits," in *Proceedings of the 25th IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[35] A. G. Rodrigo Teles Hermeto and F. Theoleyre, "Scheduling for IEEE 802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey," *Elsevier Computer Communications*, vol. 114, pp. 84–105, 2017.

[36] T. Phinney and P. T. amd RA. Assimiti, "Rpl applicability in industrial networks," IETF ROLL WG, Internet-Draft draft-ietf-roll-rpl-industrial-applicability-02 [work-in-progress], October 2013.

[37] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," IETF, RFC 6552, March 2012.

[38] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," IETF, RFC 6719, September 2012.

[39] R. Koutsiamanis, G. Papadopoulos, N. Montavont, and P. Thubert, "Common Ancestor Objective Function and Parent Set DAG Metric Container Extension," IETF ROLL WG, Internet-Draft draft-ietf-roll-nsa-extension-10 [work-in-progress], October 2020.

[40] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," IETF, RFC 6206, March 2011.

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

[42] E. Grossman, "Deterministic Networking Use Cases," IETF RFC 8578, July 2019.

[43] G. Z. Papadopoulos, P. Thubert, F. Theoleyre, and C. Bernardos, "RAW use cases," IETF RAW WG, Internet-Draft draft-ietf-raw-use-cases-01 [work-in-progress], February 2021.

[44] N. Finn, P. Thubert, B. Varga, and J. Farkas, "Deterministic Networking Architecture," IETF RFC 8655, October 2019.

[45] P. Thubert, G. Z. Papadopoulos, and R. Buddenberg, "Reliable and Available Wireless Architecture/Framework," IETF RAW WG, Internet-Draft draft-pthubert-raw-architecture-05 [work-in-progress], November 2020.

[46] N. Finn and P. Thubert, "Deterministic Networking Problem Statement," IETF RFC 8557, May 2019.

[47] P. Thubert, "Converging over deterministic networks for an Industrial Internet," Ph.D. dissertation, IMT Atlantique, 2017.

[48] "Charter IETF DetNet WG, Deterministic Networking," April 2020.

[49] N. Maeurer, T. Graeupl, and C. Schmitt, "L-band Digital Aeronautical Communications System (LDACS)," IETF, RAW WG, Internet-Draft draft-ietf-raw-ldacs-07 [work-in-progress], February 2021.

[50] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, T. Noel, and G. Schreiner, "Adding value to WSN simulation using the IoT-LAB experimental platform," in *Proceedings of the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013, pp. 485–490.

[51] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noel, "Importance of Repeatable Setups for Reproducible Experimental Results in IoT," in *Proceedings of the 13th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2016.

[52] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, "Thorough IoT testbed Characterization: from Proof-of-concept to Repeatable Experimentations," *Elsevier Computer Networks*, vol. 119, pp. 86–101, 2017.

[53] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks," in *Proceedings of the 23rd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2012, pp. 327–332.

[54] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois, "Kausa: KPI-aware Scheduling Algorithm for Multi-flow in Multi-hop IoT Networks," in *Proceedings of the 15th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now)*, 2016, pp. 47–61.

[55] N. Accettura, M. Palattella, G. Boggia, L. Grieco, and M. Dohler, "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things," in *Proceedings of the 14th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM*, 2013.

[56] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, "LLSF: Low Latency Scheduling Function for 6TiSCH Networks," in *Proceedings of the 12th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2016, pp. 93–95.

[57] I. Hosni and F. Théoleyre, "Self-Healing Distributed Scheduling for End-to-End Delay Optimization in Multihop Wireless Networks with 6TiSCH," *Computer Communications*, vol. 110, pp. 103–119, 2017.

[58] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2015, pp. 337–350.

[59] S. Oh, D. Hwang, K.-H. Kim, and K. Kim, "Escalator: An Autonomous Scheduling Scheme for Convergecast in TSCH," *Sensors*, vol. 18, no. 4, p. 1209, 2018.

[60] D. Dujovne, L. Grieco, M. Palattella, and N. Accettura, "6TiSCH Experimental Scheduling Function (SFX)," IETF 6TiSCH WG, Internet-Draft draft-ietf-6tisch-6top-sfx-01 [work-in-progress], March 2018.

[61] S. Duquennoy, X. Vilajosana, and T. Watteyne, "6TiSCH Autonomous Scheduling Function (ASF)," IETF 6TiSCH WG, Internet-Draft draft-duquennoy-6tisch-asf-01 [work-in-progress], March 2018.

[62] D. Hauweele, R.-A. Koutsiamanis, B. Quoitin, and G. Z. Papadopoulos, "Thorough Performance Evaluation & Analysis of the 6TiSCH Minimal Scheduling Function (MSF)," *Springer, Journal of Signal Processing Systems*, 2021.

[63] "Charter IETF RAW WG, Reliable and Available Wireless," January 2020.

[64] G. Z. Papadopoulos, F. Theoleyre, P. Thubert, and N. Montavont, "IETF Reliable and Available Wireless (RAW): Use Cases and Problem Statement," in *Proceedings of the 19th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now)*, 2020, pp. 303–314.

[65] F. Theoleyre, G. Z. Papadopoulos, and G. Mirsky, "Operations, Administration and Maintenance (OAM) features for RAW," IETF, RAW WG, Internet-Draft draft-theoleyre-raw-oam-support-02 [work-in-progress], April 2020.

[66] G. Fairhurst and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)," IETF, RFC 3366, August 2002.

[67] K. D. Korte, A. Sehgal, and J. Schönwälder, "A Study of the RPL Repair Process Using ContikiRPL," in *Proceedings of the 6th IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, 2012, pp. 50–61.

[68] A. Bruniaux, R.-A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, "Defragmenting the 6LoWPAN Fragmentation Landscape: A Performance Evaluation," *MDPI Sensors*, vol. 21(5), p. 1711, 2021.

[69] T. Lagos, R. A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, "Multi-path Selection in RPL based on Replication and Elimination," in *Proceedings of the 17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now)*, 2018.

[70] T. Matsui, G. Z. Papadopoulos, P. Thubert, T. Watteyne, and N. Montavont, "Poster: 4th Industrial Revolution: Toward Deterministic Wireless Industrial Networks," in *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2017.

[71] C. A. G. Da Silva and C. M. Pedroso, "Mac-layer packet loss models for wi-fi networks: A survey," *IEEE Access*, vol. 7, pp. 180 512–180 531, 2019.

[72] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "Is Local Blacklisting Relevant in Slow Channel Hopping Low-Power Wireless Networks?" in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[73] ——, "LABeL: Link-based Adaptive BLacklisting Technique for 6TiSCH Wireless Industrial Networks," in *Proceedings of the 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2017.

[74] ——, "Whitelisting without Collisions for Centralized Scheduling in Wireless Industrial Networks," *IEEE Internet of Things Journal*, vol. 6, pp. 5713–5721, 2019.

[75] V. Kotsiou, G. Z. Papadopoulos, D. Zorbas, P. Chatzimisios, and F. Theoleyre, "Blacklisting-based Channel Hopping Approaches in Low-power and Lossy Networks," *IEEE Communications Magazine*, vol. 57, pp. 48–53, 2019.

[76] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "Adaptive Multi-Channel Offset Assignment for Reliable IEEE 802.15.4 TSCH Networks," in *Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–5.

[77] M. Hänninen, J. Suhonen, T. D. Hämäläinen, and M. Hännikäinen, "Link Quality-Based Channel Selection for Resource Constrained WSNs." Springer, 2011.

[78] M. Sha, G. Hackmann, and C. Lu, "Arch: Practical channel hopping for reliable home-area sensor networks," in *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2011, pp. 305–315.

[79] 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)*, vol. 8, no. 4, p. 34, 2012.

[80] G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios, and T. Noel, "Performance Evaluation Methods in Ad Hoc and Wireless Sensor Networks: A Literature Study," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 122–128, 2016.

[81] F. Theoleyre and G. Papadopoulos, "Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2016.

[82] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, pp. 43 – 59, 2015.

[83] A. Nanda, M. P. Pai, and A. Gole, "An algorithm to automatically generate schedule for school lectures using a heuristic approach," *International Journal of Machine Learning and Computing*, vol. 2, 2012.

[84] P. H. Gomes, T. Watteyne, and B. Krishnamachari, "MABO-TSCH: Multi-hop And Blacklist-based Optimized Time Synchronized Channel Hopping," *Transactions on Emerging Telecommunications Technologies*, vol. e3223, pp. 1–20, 2017.

[85] D. J. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.

[86] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial iot applications using ieee802. 15.4 e tsch," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, 2013.

[87] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop ieee 802.15. 4e networks," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*, 2012, pp. 327–332.

[88] G. L. Gall, N. Montavont, and G. Z. Papadopoulos, "IoT Network Management within the Electric Vehicle Battery Management System," *Springer, Journal of Signal Processing Systems*, 2021.

[89] E. Municio, G. Daneels, M. Vučinić, S. Latré, J. Famaey, Y. Tanaka, K. Brun, K. Muraoka, X. Vilajosana, and T. Watteyne, "Simulating 6tisch networks," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, p. e3494, 2019.

[90] T. Chang, M. Vučinić, X. Vilajosana, D. Dujovne, and T. Watteyne, "6TiSCH Minimal Scheduling Function: Performance Evaluation," *Internet Technology Letters*, vol. 3, 05 2020.

[91] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. Pister, "A realistic energy consumption model for tsch networks," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482–489, 2014.

[92] H.-P. Le, M. John, and K. Pister, "Energy-aware routing in wireless sensor networks with adaptive energy-slope control," *EE290Q-2 Spring*, 2009.

[93] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, "LLSF: Low latency scheduling function for 6TiSCH networks," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2016, pp. 93–95.

[94] I. Hosni, F. Théoleyre, and N. Hamdi, "Localized scheduling for end-to-end delay constrained low power lossy networks with 6tisch," in *IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 507–512.

[95] S. C. F. A. Rincon Vija, G. Z. Papadopoulos, and N. Montavont, "Enabling Robust Wireless Communication for BMS on Electric Vehicles," in *Proceedings of the 46th IEEE Conference on Local Computer Networks (LCN 2021)*, 2021.

[96] M. J. Best and K. Ritter, *Linear Programming Active Set Analysis and Computer Programs*. Prentice Hall Upper Saddle River, NJ, 1985.

[97] N. R. Sabar, L. M. Kieu, E. Chung, T. Tsubota, and P. E. Maciel de Almeida, "A Memetic Algorithm for Real World Multi-intersection Traffic Signal Optimisation Problems," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 45–53, 2017.

[98] G. L. Gall, "Wireless Network for Reliable Electric Vehicle Battery Management," Ph.D. dissertation, IMT Atlantique, 2021.

[99] G.-W. Lee and E.-N. Huh, "Reliable data transfer using overhearing for implicit ack," in *Proceedings of the International Joint Conference on ICCAS-SICE*, 2009, pp. 1976–1979.

[100] P. Minet, I. Khoufi, and A. Laouiti, "Increasing reliability of a TSCH network for the industry 4.0," in *Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA)*, 2017.

[101] T. L. Jenschke, "Toward Reliable and Bounded Latency for Internet of Things," Ph.D. dissertation, IMT Atlantique, 2020.

[102] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*, 2004, pp. 455–462.

[103] R.-A. Koutsiamanis, G. Z. Papadopoulos, B. Quoitin, and N. Montavont, "A Centralized Controller for Reliable and Available Wireless Schedules in Industrial Networks," in *Proceedings of the 16th International Conference on Mobility, Sensing and Networking (MSN)*, 2020.

[104] S. Bera, S. Misra, and A. V. Vasilakos, "Software-Defined Networking for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[105] P. Di Dio, S. Faraci, L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, and P. Livreri, "Exploiting state information to support QoS in Software-Defined WSNs," in *Proceedings of the 15th IFIP Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2016.

[106] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 513–521.

[107] R. C. A. Alves, D. A. G. Oliveira, G. A. Nunez Segura, and C. B. Margi, "The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks," *IEEE Access*, vol. 7, pp. 115 093–115 108, 2019.

[108] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Towards an SDR implementation of LoRa: reverse-engineering, demodulation strategies and assessment over Rayleigh channel," *Elsevier Computer Communications*, vol. 153, pp. 595–605, 2020.

[109] ——, "Investigating Theoretical Performance and Demodulation Techniques for LoRa," in *Proceedings of the 1st International Workshop on Data Distribution in Industrial and Pervasive Internet (DIPI)*, 2019, pp. 1–6.

[110] T. Theodorou and L. Mamatas, "A Versatile Out-of-Band Software-Defined Networking Solution for the Internet of Things," *IEEE Access*, vol. 8, pp. 103 710–103 733, 2020.

[111] 3GPP, "Study on New Radio Access Technology: Radio Access Architectures and Interfaces. 3GPP Technical Specification," 3GPP, Technical Specification, 2018.

[112] G. Americas, "Innovations in 5G Backhaul Technologies)," https://www.5gamericas.org/innovations-in-5g-backhaul-technologies/, June 2020, accessed: April 2021.

[113] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[114] J. S. Lu, D. Steinbach, P. Cabrol, and P. Pietraski, "Modeling Human Blockers in Millimeter Wave Radio Links," *ZTE Communications*, vol. 10, no. 4, pp. 23–28, 2012.

[115] 3GPP, "R1-1812199, System Performance Evaluation in Multi-Hop IAB network," Nov 2018.

[116] M. Polese, M. Giordani, A. Roy, D. Castor, and M. Zorzi, "Distributed Path Selection Strategies for Integrated Access and Backhaul at mmWaves," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[117] 3GPP, "RP-193251, Enhancements of Integrated Access and Backhaul," Dec 2019.

[118] R. E. Navas, "Improving the resilience of the constrained Internet of Things: a moving target defense approach," Ph.D. dissertation, IMT Atlantique, 2020.

[119] R. E. Navas, F. Cuppens, N. B. Cuppens, L. Toutain, and G. Z. Papadopoulos, "MTD, Where Art Thou? A Systematic Review of Moving Target Defense Techniques for IoT," *IEEE Internet of Things Journal*, 2020.

[120] R. Navas, H. Sandaker, F. Cuppens, N. Cuppens-Boulahia, L. Toutain, and G. Z. Papadopoulos, "IANVS: A Moving Target Defense Framework for Internet of Things," in *Proceedings of the 25th IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[121] R. E. Navas, F. Cuppens, N. B. Cuppens, L. Toutain, and G. Z. Papadopoulos, "Physical resilience to insider attacks in IoT networks: Independent cryptographically secure sequences for DSSS anti-jamming," *Elsevier Computer Networks*, vol. 187, p. 107751, 2021.

A full list of my publications is available at
`http://www.georgiospapadopoulos.com`