ISEP

Instituto Superior de Engenharia do Porto

Fast mobility support in low-power wireless networks

smart-HOP over RPL/6LoWPAN

Master Thesis

To obtain the degree of master at the
Instituto Superior de Engenharia do Porto,
public defend on July 11th 2013 by

Daniel Augusto da Rocha Moreira

Master in Electrical and Computer Engineering
Telecommunications Specialization
Porto, Portugal.

Supervisor: Mário Jorge de Andrade Ferreira Alves (PhD)

Co-Supervisor: Hossein Fotouhi

Composition of Supervisory Committee: José António Tenreiro Machado (PhD)
Mário Jorge de Andrade Ferreira Alves (PhD)
Paulo José Lopes Machado Portugal (PhD)

Author email: 1050749@isep.ipp.pt

To my parents and brother.

Acknowledgments

I would like to express the deepest appreciation to Hossein Fotouhi. Without whom this thesis wouldn't be possible. To Professor Mário Alves, for the opportunity presented and excellent guidance.

I thank my family not only for the support during this research, but also throughout my academic career. To Marco Otero and Ricardo Moreira, for the constant encouragement and friendship.

Last but not least, to Inês de Castro for the support, patience and caring.

Abstract

With the emergence of low-power wireless hardware new ways of communication were needed. In order to standardize the communication between these low powered devices the Internet Engineering Task Force (IETF) released the 6LoWPAN standard that acts as an additional layer for making the IPv6 link layer suitable for the lower-power and lossy networks. In the same way, IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) has been proposed by the IETF Routing Over Low power and Lossy networks (ROLL) Working Group as a standard routing protocol for IPv6 routing in low-power wireless sensor networks. The research performed in this thesis uses these technologies to implement a mobility process.

Mobility management is a fundamental yet challenging area in low-power wireless networks. There are applications that require mobile nodes to exchange data with a fixed infrastructure with quality-of-service guarantees. A prime example of these applications is the monitoring of patients in real-time. In these scenarios, broadcasting data to all access points (APs) within range may not be a valid option due to the energy consumption, data storage and complexity requirements. An alternative and efficient option is to allow mobile nodes to perform hand-offs.

Hand-off mechanisms have been well studied in cellular and ad-hoc networks. However, low-power wireless networks pose a new set of challenges. On one hand, simpler radios and constrained resources ask for simpler hand-off schemes. On the other hand, the shorter coverage and higher variability of low-power links require a careful tuning of the hand-off parameters.

In this work, we tackle the problem of integrating smart-HOP within a standard protocol, specifically RPL. The simulation results in Cooja indicate that the proposed scheme minimizes the hand-off delay and the total network overhead. The standard RPL protocol is simply unable to provide a reliable mobility support similar to other COTS technologies. Instead, they support joining and leaving of nodes, with very low responsiveness in the existence of physical mobility.

Contents

List of Figures

List of Tables

Acronyms

AODV	Ad hoc On-Demand Distance Vector Routing
AP	Access Point
API	Application Programming Interface
CDMA	Code Division Multiple Access
DAG	Directed Acyclic Graph
DAO	DODAG Destination Advertisement Object
DAO-ACK	DODAG Destination Advertisement Object Acknowledge
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Oriented Directed Acyclic Graph
DTSN	Destination Advertisement Trigger Sequence Number
ETT	Expected Transmission Time
ETX	Expected Transmission Count
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IID	Interface Identifier
IS-IS	Intermediate System to Intermediate System

LLN	Low Power and Lossy Networks
LQE	Link Quality Estimator
LR-WPAN	Low-Rate Wireless Personal Area Network
MN	Mobile Node
MOP	Mode of Operation
ND	Neighbor Discovery
OCP	Objective Code Point
OF	Objective Function
OF0	Objective Function 0
OLSR	Optimized Link State Routing Protocol
OSPF	Open Shortest Path First
PDR	Packet Delivery Ratio
PRR	Packet Reception Ratio
ROLL	Routing Over Low power and Lossy networks
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
RSSI	Received Signal Strength Indication
SAA	Stateless Address Autoconfiguration
SNMP	Simple Network Management Protocol
SNR	Signal-to-Noise Ratio
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XML	eXtensible Markup Language

1
Overview

1.1 Research context

Nowadays, mobility is one of the major requirements in several emerging ubiquitous and pervasive sensor network applications, including health-care monitoring, intelligent transportation systems and industrial automation [6, 7, 8]. In some of these scenarios, mobile nodes are required to transmit data to a fixed-node infrastructure in a timely and reliable fashion. For example, in clinical health monitoring [9, 10], patients embed wireless sensing devices that report data through a fixed wireless network infrastructure. In these type of scenarios, it is necessary to provide a reliable and constant stream of information.

Mobility management is a wide area that covers various aspects such as hand-off process, re-routing, re-addressing and security issues. In this research, our main focus is on enabling mobility support within commercial and standard low-power wireless network protocols. In this way, we are aiming to integrate smart-HOP within an existing standard routing algorithm. smart-HOP is a hand-off process tailored for wireless sensor networks. *Hand-off* refers to the process in which a mobile node disconnects from a serving point of attachment and attaches itself to a new point of attachment.

Mobility in general can alternatively be described in terms of micro-mobility and macro-mobility. *Micro-mobility* refers to the case where the node moves within a network domain. *Macro-mobility* on the other hand refers to the mobility between networks [1]. In this work, we tackle the hand-off process within the micro-mobility context.

Link Quality Estimator (LQE) is one of the main challenges in a hand-off process. The dynamic changes of low-power links require an accurate and fast estimation of the link. Selecting a proper link estimation needs studying the characteristics of dynamic, unreliable and variable wireless links in the existence of mobility.

In mobile wireless sensor network applications, a good link quality metric is essential to a reliable and energy-efficient system operation. However, harsh environments with dynamics, rapid variations of wireless channel preclude an efficient mechanism for knowing instantaneous link quality at the time of transmission, thus making it difficult to estimate the instantaneous value of the wireless link quality.

Most of link quality metrics combine a number of parameters to estimate the status of the link. They declare that the Received Signal Strength Indication (RSSI) or Signal-to-Noise Ratio (SNR) indicators are not suitable for determining the quality of wireless links [11]. However, we argue that these heuristics are more recommended and practical for networks with static nodes with less variability of wireless links. This statement was also confirmed in some other works [12, 13, 14]. In mobile networks with dynamic topology, wireless links are highly unreliable and variable. The sophisticated link metrics require high processing and responsiveness while the node is moving. In fact, in critical applications with timely demands, the multi-criteria hand-off decisions lose the responsiveness and accuracy. This is the main reason of leading these networks to benefit from fast hand-off decisions by relying on the RSSI/SNR values and fine tuning the related parameters.

Motivation. A naive solution in these applications would be for mobile nodes to broadcast the information to all Access Points (AP) within range. The APs are the static nodes that build the infrastructure of the network. The broadcast approach, while simple, has a major limitation. Broadcasts lead to redundant information at neighboring APs (since several of them receive the same packets). This implies that the fixed infrastructure has to either waste resources in forwarding the same information to the end point, or it needs a complex scheme, such as data fusion, to eliminate duplicated packets locally.

A more efficient solution is for mobile nodes to use a single AP to transmit data at any given time. This alternative would require nodes to perform reliable and fast hand-offs between neighboring APs. Hand-offs have been studied extensively in other wireless systems [15, 16, 17, 18, 19, 20, 21, 22], in particular cellular and WLAN networks. However, these techniques are not suitable for Wireless Sensor Networks (WSN) due to their characteristics. Contrary to more powerful systems, such as cellular networks, which have advanced spread spectrum radios and almost unlimited energy resources, WSNs typically have severely constrained resources.

1.2 Problem Statement

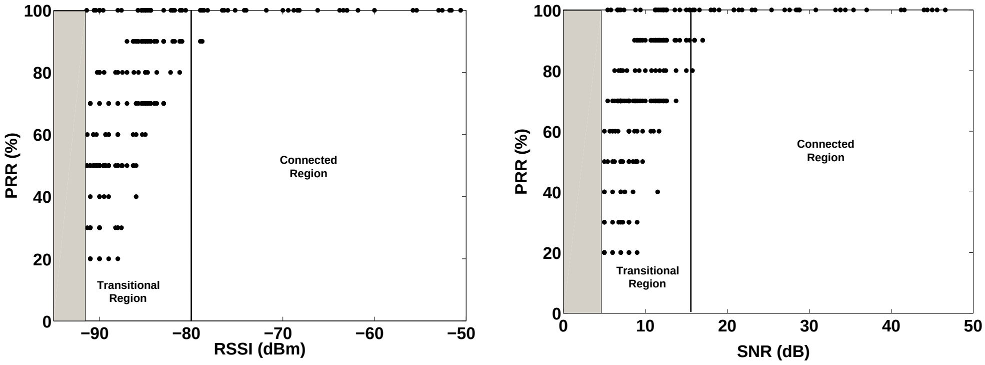
As we stated earlier, this thesis addresses the integration of smart-HOP process within the standard protocols in low-power applications. In this way, there are some challenges that should be carefully considered.

Low-power links. Wireless links in sensor networks have two characteristics that affect the hand-off process: short coverage and high variability [23]. Short coverage imply low densities of access points. In cellular networks, for example, it is common to be within the range of tens of APs. This permits the node to be conservative with thresholds and to select links with very high reliability. On the other hand, sensor networks may not be deployed in such high densities, and hence, the hand-off should relax its link quality requirements. In practice, this implies that the hand-off parameters should be more carefully calibrated within the (unreliable) transitional region.

The high variability of links has an impact in stability. When not designed properly, hand-off mechanisms may degrade the network performance due to the *ping-pong* effect, which consists in mobile nodes having consecutive and redundant hand-offs between two APs due to sudden fluctuation of their link qualities. This happens usually when a mobile node moves in the frontiers of two APs. Hence, to be stable, a hand-off mechanism should calibrate the appropriate thresholds according to the particular variance of its wireless links.

The variation of RSSI and SNR parameters gives a good resolution on the low-power link characteristics in low-power links. In the sensor networks community, the de-facto way to classify links is to use the connected, transitional and disconnected regions. Figure 1.1 depicts these three regions which agree with the previous studies [10, 24]. The SNR is calculated by measuring the noise floor immediately after receiving the packet, and then, subtracting it from the RSSI value. The RSSI regions can be mapped directly to the SNR ones by subtracting the average noise floor.

The transitional region in sensor networks, for the CC2420 radio transceiver, encompasses the approximate range [-92 dBm, -80 dBm] (shown in Figure 1.1). Intuition may dictate that the closer the hand-off is performed to the connected region the better (because links are more reliable). In practice, a hand-off starts when the link with the current (serving) AP drops below a given value (TH_{low})



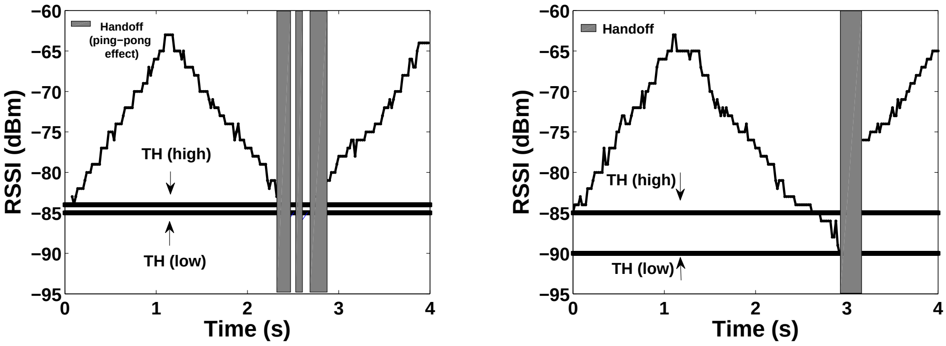
(a) (b)

Figure 1.1: *Low-power link model (a) RSSI vs. PRR For RSSI greater than -80 dBm, the PRR is greater than 90%, and for RSSI less than -92 dBm, the PRR is less than 10%. In between, a small variation in the RSSI can cause a big difference in the PRR, which is identified as transitional region. (b) SNR vs. PRR. The borders for SNR are 4 dB and 16 dB, which are obtained by subtracting the noise floor from the RSSI readings [10].*

and stops when it finds a new AP with the required link quality (above TH_{high}). Figure 1.2(a) depicts this conservative approach. It considers -85 dBm as the lower threshold, and the upper threshold is 1 dBm higher. These parameters lead to a negative effect: a long delay (≈0.7 s) that takes three hand-offs between the two contiguous APs (ping-pong effect). Figure 1.2(b) shows that by considering a wider margin, deeper into the transitional region, the ping-pong effect disappears and the delay is reduced to approximately 0.2 s.

Hard or soft hand-off for WSNs. The type of hand-off is dictated by the capabilities of the radio, standards and technologies. Hand-offs are classified into two main categories: hard hand-offs and soft hand-offs. In a *soft hand-off*, the radio can use multiple channels at the same time. This characteristic enables a mobile node to communicate with several APs and assess their link qualities while transmitting data to the serving AP. A common technology used in soft hand-off radios is Code Division Multiple Access (CDMA) [25].

In a *hard hand-off*, the radio can use only one channel at any given time, and hence, it needs to stop the data transmission before the hand-off process starts. Consequently, in hard hand-offs it is central to minimize the time spent looking for a new AP. WSN nodes typically rely on low-power radio transceivers that can operate on a single channel at a time, such as the widely used CC2420. This implies that current WSN should utilize a hard hand-off approach.



(a) (b)

Figure 1.2: *(a) an example of an inefficient hand-off with narrow hysteresis margin (1 dBm), $TH_{low} = -86$ dBm and $TH_{high} = -85$ dBm. (b) an example of an efficient hand-off with wide hysteresis margin (5 dBm), $TH_{low} = -90$ dBm and $TH_{high} = -85$ dBm [10].*

1.3 Research Objectives

The primary objective of this research is achieving reliable and real-time micro-mobility support in low-power wireless networks. To reach this primary objective, a range of scientific and technical objectives have been investigated.

- Devise an efficient algorithm that integrates smart-HOP within RPL routing.
- Implement that algorithm in one of the COTS Operating Systems.
- Compare the results of smart-HOP performance with RPL algorithm in terms of hand-off delay and network overhead.

1.4 Research contribution

1. *Connectivity.* This is a process that tracks the reachability of a child to its parent. In low data rate applications, it is more likely to lose the connectivity between a child and a parent due to the link degradation or moving one party.
2. *Mobility detection.* We managed a continuous link quality observation to detect the exact moment of movement.
3. *Parent selection.* We employed smart-HOP algorithm to select the best parent by fine tuning the relevant parameters.

1.5 Structure of the thesis

This thesis is organized as follows. Section 2 describes the core technologies and tools used in this work, as well as a general overview of the mobility research and its relevance to a wide set of applications.

In Section 3, RPL is addressed. The routing protocol used and modified by our work is described in a detailed manner.

Section 4 depicts all the information related to the smart-HOP algorithm. The devising of the algorithm, model analysis and observations that were used to implement it within RPL and Contiki.

Section 5 presents the integration of smart-HOP within RPL. The algorithm developed, methodologies and simulation results are described here.

The last section corresponds to the conclusion. Main topics are discussed and future work is presented, finalising the manuscript.

2
Background

The Internet has been a great success over the past 20 years, growing from a small academic network into a global ubiquitous network used regularly by nearly 2 billion people. As the Internet of routers, servers and personal computers has been maturing, another Internet revolution has been going on — *The Internet of Things.* The Internet of Things (IoT) is a computing concept that describes a future where everyday physical objects will be connected to the Internet and will be able to identify themselves to other devices. The impact of the Internet of Things will be significant, with the promise of better environmental monitoring, energy savings, smart grids, better logistics, better healthcare and smart homes [26, 27, 28, 29, 30, 31].

Rapid growth of embedded control and monitoring systems in almost any electronic device and the need for connectivity of these applications is causing an integration bottleneck. Conventionally, these communication links were wired. Wires allow power and a reliable transmission of signals from a controller to its peripherals. When the peripherals are not physically contained in the controller, the required wiring brings issues such as cost of installation, safety, and operation convenience to the surface. Wireless technology is a solution to overcome these obstacles, although it comes with its own set of challenges such as propagation, interface, security and connectivity. The technology to overcome these issues exists, but normally with added complexity causing an increase in the cost of the system. Among various wireless technologies Low-Rate Wireless Personal Area Network (LR-WPAN) is specificly designed for low-cost, low-power and short-range wireless communications.

2.1 Low-power wireless networks

Wireless Sensor Network (WSN) are a subset of wireless networking applications focused on enabling connectivity between wireless sensors and actuators. IEEE 802.15.4 Working Group is chartered to focus on wireless sensor networks. WSNs share most of the issues surrounding wireless applications such as information security, authentication, small-scale radio-frequency propagation and antenna placement. Mobility is a benefit of wireless solution, although in the WSN context, this capability is traded with ease of installation [32]. In other words, mobility is normally not a requirement for a WSN system, but certain mobility concepts can be used to enable ad-hoc networking. It is important to clarify that the term mobility in this context refers to relative motion of devices with respect to each other (physical mobility). The set of advantages described is not enough to replace hardwired connections. The reliability and security (perceived and real) of wired networks can be higher than the wireless communication systems.

It is expected, however, that hybrid networks, wired and wireless, will coexist. Wireless sensors will act as extensions of wired networks wherever the wireless capability adds value to the specific application. The inertia slowing the widespread implementation of WSNs is the lack of standardized technologies that can address their requirements both at the application level and from the communications point of view. The focus of the wireless industry has been primarily on communications with higher data throughput, leaving short-range wireless connectivity behind.

WSN nodes have several restrictions, e.g., limited energy supply, limited computing power, and limited bandwidth of the wireless links connecting sensor nodes. One of the main design goals of WSNs is to carry out data communication while trying to prolong the lifetime of the network and prevent connectivity degradation by employing aggressive energy management techniques. The design of routing protocols in WSNs is influenced by many challenging factors. These factors must be overcome before efficient communication can be achieved in WSNs. Key features which make WSNs applicable in these domains —and even preferable to conservative deployments in which clients report data directly to a centralized access point— are the low cost of the single sensor devices, which make a deployment feasible, as well as their un-intrusiveness in terms of size and radiation.

Two sorts of deployments are distinguished in WSNs: nodes can be distributed in a structured way, by putting them in pre-planned positions. However, in some cases this might not be possible, as, e.g., the terrain which is supposed to be monitored is intoxicated and would endanger the engineers deploying the network. In such circumstances, WSNs allow for an unstructured deployment where nodes are randomly distributed over the desired area. Unstructured deployments tend to need a higher density of nodes than structured ones, as their placement can not be optimized to achieve a good coverage of the environment [33].

Nodes usually run on batteries and are therefore limited in their power resources. This imposes a hard limit on the lifespan of a WSN. Since radio communication is the most expensive action a node performs [34], communication protocols for WSNs need to minimize the number of times a node needs to communicate. In the next section, we refer some of these communication protocols that had to be developed to cope with the needs of resource constrained devices.

2.2 Standard and COTS technologies

Several standards are currently either ratified or under development for wireless sensor networks.

- **WirelessHART** is an extension of the HART[1] Protocol and is specifically designed for industrial applications like process monitoring and control.
- **ZigBee technology** is a low data rate, low-power consumption, low-cost, wireless networking protocol targeted towards automation and remote control applications. IEEE 802.15.4 committee started working on a low data rate standard a short while later. Then the ZigBee Alliance and the IEEE decided to join forces and ZigBee is the commercial name for this technology.
- **6LoWPAN** is the IETF standards track specification for the IP-to-MAC-Layer mapping for IPv6 on IEEE 802.15.4.

WirelessHART devices communicate using Time Division Multiple Access (TDMA). Each WirelessHART device maintains a precise sense of time and remains synchronized with all neighbouring devices. All device-to-device communication is done in a pre-scheduled time-window that enables very reliable (collision-free), power-efficient, and scalable communication. ZigBee, WirelessHART, and 6lowpan all are based on the same underlying radio standard: IEEE 802.15.4. In Table 2.1, we represent the characteristics of these communication protocols.

[1] Highway Addressable Remote Transducer is a protocol used in real time communication systems. It is one of the most popular industrial protocols today. Developed by Rosemount Inc., it was made an open protocol in 1986. Since then, the capabilities of the protocol have been enhanced by successive revisions to the specification.

Table 2.1: *WSN Technologies [5]*

Standard	**ZigBee**	**6LoWPAN**	**WrelessHART**
Main application	Control and monitoring	Control and monitoring	Industrial control and monitoring
Memory	4-32 kB	4-32 kB	
Battery Lifetime (days)	100-1000+	100-365+	760+
Network nodes	255	65536	200
Throughput	Up to 250 Kbps	Up to 250 Kbps	Up to 250 Kbps
Range	1-75	1-100	1-100
Main feature	Reliability, low consume, low cost	IPv6 over IEEE 802.15.4	Reliability

2.2.1 IEEE 802.15.4

IEEE 802.15.4 is a standard protocol that specifies the physical layer and media access control (MAC) for low-rate wireless personal area networks (LR-WPANs). It is maintained by the IEEE 802.15 working group and the first version was completed in May 2003. The IEEE 802.15.4 standard specifies a wireless interface meant for wireless embedded applications, such as building automation, industrial automation and other sensing and tracking purposes. The standard is very flexible, allowing from ad-hoc mesh networks to infrastructure based tree topologies. The IEEE 802.15.4 is the basis for the ZigBee networking stack and WirelessHART, each of which further attempts to offer a complete networking solution by developing the upper layers (which are not covered by the standard). Alternatively, it can be used with 6LoWPAN and standard Internet protocols [5]. In Table 2.2, the main features of IEEE 802.15.4 are presented.

Table 2.2: *IEEE 802.15.4 features [5]*

Frequency bands and data rates	868-868.8 MHz and 20 Kb/s 902-928 MHz and 40 Kb/s 2400-2483.5 MHz and 250 Kb/s
Range	10-20 m
Addressing	IEEE 64-bit addresses
Network nodes	Up to 2^{64}
Security	128 AES
Channel access	CSMA-CA

The features of the PHY (physical layer) are activation and deactivation of the radio transceiver, ED (Energy Detection), LQI (Link quality Indication), channel selection, clear channel assessment (CCA), and transmitting as well as receiving packets across the physical medium. The radio operates at one or more of the following bands.

- 868-868.8 MHz: Europe, allows one communication channel
- 902-928 MHz: North America, up to ten channels, extended to thirty in 2006 revision
- 2400-2483.5 MHz: worldwide use, up to sixteen channels

As mentioned in Table2.2, the 2.4 GHz physical layer provides a data rate up to 250 kbps, but lower rates can be considered using different frequency bands. In 2.1 is represented the channel distribution of a IEEE 802.15.4 communication.

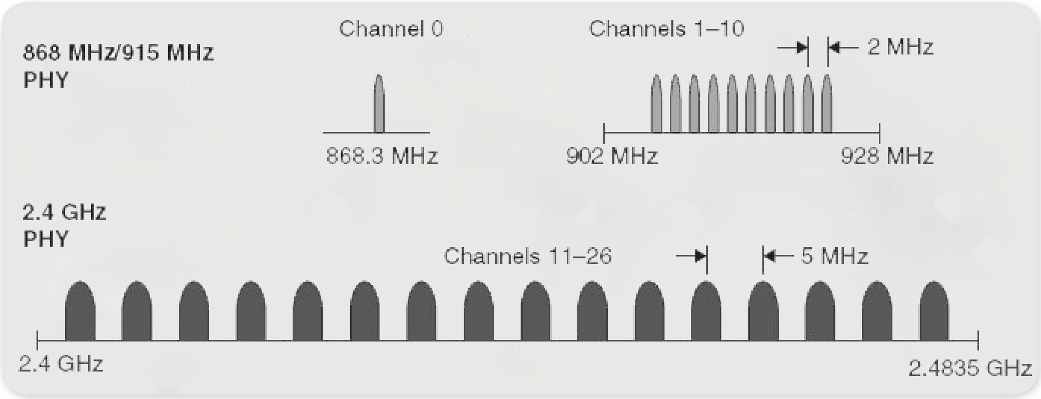


Figure 2.1: *IEEE 802.15.4 channel distribution*

While any of these bands can technically be used by 802.15.4 devices, the 2.4 GHz band is more popular as it is open in most of the countries worldwide. General Concern exist around interference in 2.4 GHz space with devices such as WiFi, Microwave Ovens, cordless phones, wireless video systems, etc. 802.15.4 was designed from ground up with co-existence in mind. Consider a placement where various wireless networks can be present working at the same frequency bands. It is necessary to implement a dynamic selection of channels.

- MAC layer includes searching algorithms to find the best channel through the list of the possible ones.

- PHY layer implements some functions to detect the received energy, consider the quality of the channel and channel commutation.

In the next figure it is shown how wireless sensor networks can coexist with a WiFi (802.11) network without interfering. The channels that can be used in IEEE 802.15.4 at 2.4 GHz are: 15, 20, 25 and 26.

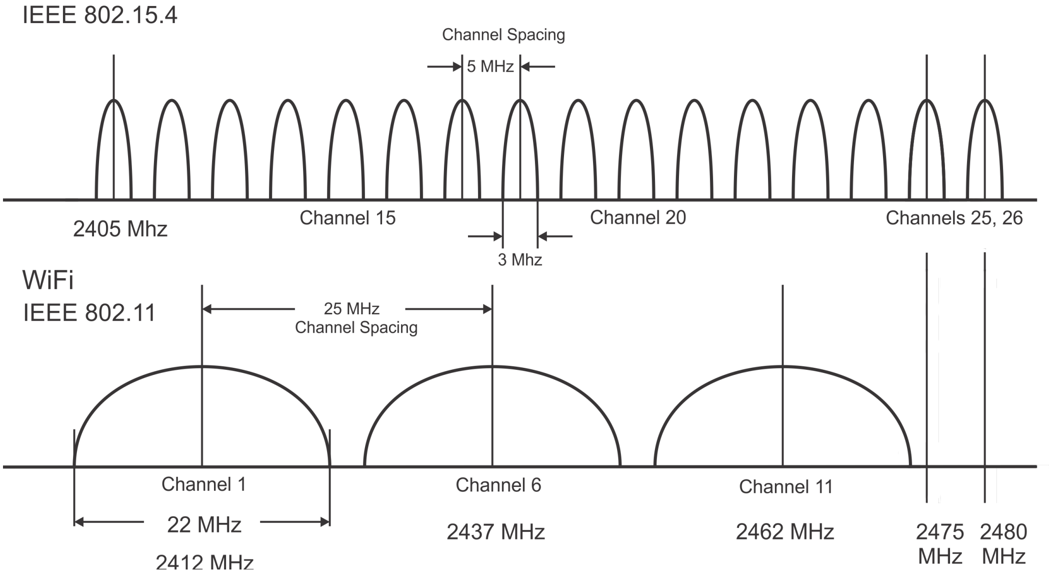


Figure 2.2: *IEEE 802.15.4 and IEEE 802.11 channels*

The 2.4 GHz employs a 16-ary quasi-orthogonal modulation technique based on DSSS. Binary data is grouped into 4-bit symbols, each symbol specifying one of 16 nearly orthogonal 32-bit chip pseudo noise (PN) sequences for transmission. PN sequences for successive data symbols are concatenated and the aggregate chip is modulated onto the carrier using minimum shift keying (MSK). The use of nearly orthogonal symbol sets simplifies the implementation, but incurs minor performance degradation. In terms of energy conservation, orthogonal signalling performs better than differential BPSK. However, in terms of receiver sensitivity, the 868/915 MHz layer has a 6-8 dB advantage [35]. Modulation parameters are summarized in the following table.

IEEE 802.15.4 has a maximum physical layer packet of 127 bytes and MAC Layer of 102 octets. MAC layer supports security mechanisms, which in the extreme case are AES.CCM-128 based, imposing an overhead of 21 octets, leaving only 81 octets for data packets. IEEE 802.15.4 supports also two MAC addresses, 16-bit short and IEEE 64-bit extended, and as mentioned before, the biggest characteristic is its low bandwidth, starting with 20 kbps at 868 MHz, 40 Kbps at 915 MHz, and at moment

Table 2.3: *IEEE 802.15.4 Modulation characteristics [5]*

Bandwidth	Chip rate Kchip/s	Modulation	Bit rate Kb/s	Symbol rate Ksymbol/s	Symbols
868-868.6	300	BPSK	20	20	Binary
902-928	600	BPSK	40	40	Binary
2400-2483.5	2000	O-QPSK	250 62.5	16 orthogonal	

reaching the 250 Kbps at 2.4GHz. Thus, over IEEE 802.15.4 to perform a reliable Personal Area Network, there are different entities, or different nodes.

All 802.15.4 networks have one unique PAN coordinator, a node responsible for all networks, being the interface with the exterior, so-called sink node in WSNs. Other two different devices constitute the PAN, the reduced-function devices (RFD) and the full-function devices (FFD). FFDs are devices with more powerful capabilities than RFDs, having the ability to work as a router, indispensable function in mesh topologies. RFDs are only end nodes, with limited capabilities. In WSNs FFDs can work as PAN coordinators (sink node), coordinators or as end-nodes. RFDs can just work as end-nodes [36]. These different devices allow the constitution of the star, tree and mesh topologies represented in the next figure.

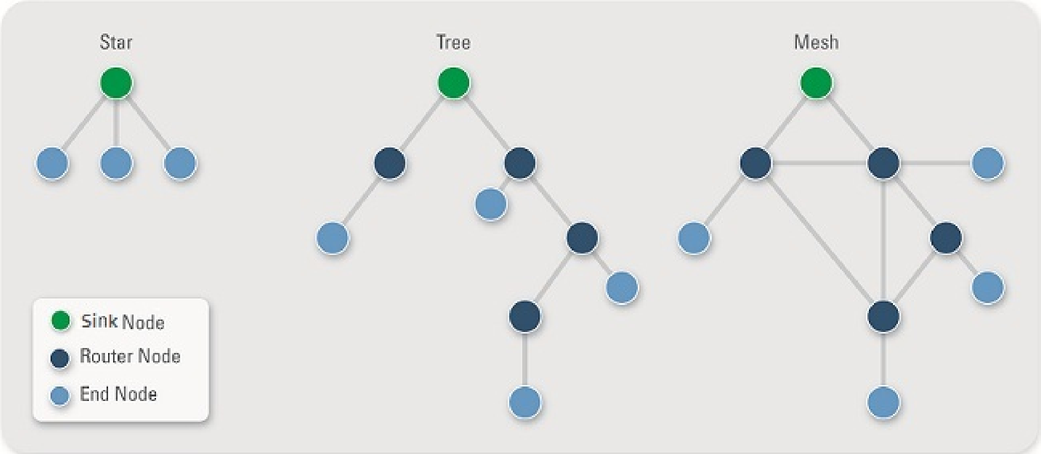


Figure 2.3: *Star, Tree and Mesh topologies in WSNs*

Understanding different network topologies will also aid in determining which protocol to select as well as where to place measurement nodes and routers. IEEE 802.11 systems are typically configured in a star or tree topology with a central or distributed access point(s) and clients 30 to 100 m from the access point depending on the wireless environment. While standard Wi-Fi installations support repeaters or routers to extend distance with a tree topology, they do not support mesh networking. Mesh networking is the ability for a node or device to route packets through multiple paths back to the gateway, and is supported by communication protocols such as ZigBee and WirelessHART, which are based on IEEE 802.15.4. Mesh networking can add distance and reliability to your wireless sensor network, but it also increases the complexity and power consumption.

2.2.2 6LoWPAN

IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) is a simple low-cost communication protocol that allows wireless connectivity in applications with limited power. 6LoWPAN adopts the IPv6 protocol stack for seamless connectivity between IEEE 802.15.4 based networks and the IPv6-based infrastructure. This section gives an overview of LoWPANs and describes how they benefit from IP and, in particular, IPv6 networking. It describes LoWPAN requirements with regards to the IP layer and the above, and spells out the underlying assumptions of IP for LoWPANs.

Why 6LoWPAN? There are a huge range of applications that could benefit from a Wireless Embedded Internet approach. Today these applications are implemented using a wide range of proprietary technologies which are difficult to integrate into larger networks and with Internet-based services. The benefits of using Internet protocols in these applications, and thus integrating them with the Internet of Things include.

- IP-based device can be connected easily to other IP networks without the need for translation gateways or proxies.
- IP networks allow the use of existing network infrastructure.
- IP-base technologies have existed for decades, are very well known, and have been proven to work and scale. The *socket* Application Programming Interface (API) is one of the most well-known and widely used APIs in the world.
- IP technology is specified in an open and free way, with standards processes and documents available to anyone. The result is that IP technology encourages innovation and is better understood by a wider audience.
- Tools for managing, commissioning and diagnosing IP-based networks already exist (although many management protocols need optimization for direct use with 6LoWPAN Nodes)

Until 6LoWPAN's development only powerful embedded devices and networks have been able to participate natively with the Internet. Direct communication with traditional IP networks requires many Internet protocols, often requiring an operating system to deal with the complexity and maintainability. Traditional Internet protocols are demanding for embedded devices for the following reasons.

- **Security**. IPv6 includes optional support for IP Security (IPsec) [37] authentication and encyption, and web services typically make use of secure sockets or transport layer security mechanisms. These techniques may be too complex, especially for simple embedded devices.
- **Web services**. Internet services today rely on web-services, mainly using the Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP), Simple Object Access Protocol (SOAP) and eXtensible Markup Language (XML) with complex transaction patterns.
- **Management**. Management with the Simple Network Management Protocol (SNMP) and web-services is often inefficient and complex.
- **Frame size**. Current Internet protocols require links with sufficient frame length (minimum of 1280 bytes for IPv6), and heavy application protocols require substantial bandwidth.

These requirements have in practice limited the Internet of Things to devices with a powerful processor, an operating system with a full TCP/IP stack, and an IP-capable communication link. A large majority of embedded applications involve limited devices, with low-power wireless and wired network communications. Wireless embedded devices and networks are particularly challenging for Internet protocols.

- **Power and duty-cycle**. Battery-powered wireless devices need to keep low duty cycles (the percentage of time active). The basic assumption of IP is that a device is always connected.
- **Multicast**. Wireless embedded radio technologies, such as IEEE 802.15.4, do not typically support multicast, and flooding in such a network is wasteful of power and bandwidth. Multicast is crucial to the operation of many IPv6 features.
- **Mesh topologies**. The applications of wireless embedded radio technology typically benefit from multihop mesh networking to achieve the required cov-

erage and cost efficiency. Current IP routing solutions may not easily be applicable to such networks.

- **Bandwidth and frame size.** Low-power wireless embedded radio technology usually has limited bandwidth (on the order of 20-250 kbit/s) and frame size (on the order of 40-200 bytes). In mesh topologies, bandwidth further decreases as the channel is shared and is quickly reduced by multihop forwarding. The IEEE 802.15.4 standard has a 127-byte frame size, with layer-2 payload sizes as low as 72 bytes. The minimum frame size for standard IPv6 is 1280 bytes [38], thus requiring fragmentation.

- **Reliability**. Standard Internet protocols are not optimized for low-power wireless networks. For example, TCP is not able to distinguish between packets dropped because of congestion or packets lost on wireless links. Further unreliability occurs in wireless embedded networks because of node failure, energy exhaustion and sleep duty cycles.

The IETF 6LoWPAN working group was created to tackle these problems, and to specifically enable IPv6 to be used with wireless embedded devices and networks. Features of the IPv6 design such as a simple header structure, and its hierarchical addressing model, made it ideal for use in wireless embedded networks with 6LoWPAN. Additionally, by creating a dedicated group of standards for these networks, the minimum requirements for implementing a lightweight IPv6 stack with 6LoWPAN could be aligned with the most minimal devices.

By designing a version of Neighbor Discovery (ND) specifically for 6LoWPAN, the particular characteristics of low-power wireless mesh networks could be taken into account. The result of 6LoWPAN is the efficient extension of IPv6 into the wireless embedded domain, thus enabling end-to-end IP networking and features for a wide range of embedded applications. Refer to RFC4919 [39] for the detailed assumptions, problem statement and goals of early 6LoWPAN standardization. Although 6LoWPAN was targeted originally at IEEE 802.15.4 radio standards and assumed layer-2 mesh forwarding [2], it was later generalized for all similar link technologies, with additional support for IP routing in RFC6775 [40].

The Protocol Stack. Figure 2.4 shows the IPv6 protocol stack with 6LoWPAN in comparison with a typical IP protocol stack and the corresponding five layers of the Internet Model. The Internet Model is sometimes referred to as a *narrow waist* model, as the Internet Protocol ties together a wide variety of link-layer technologies with multiple transport and application protocols.

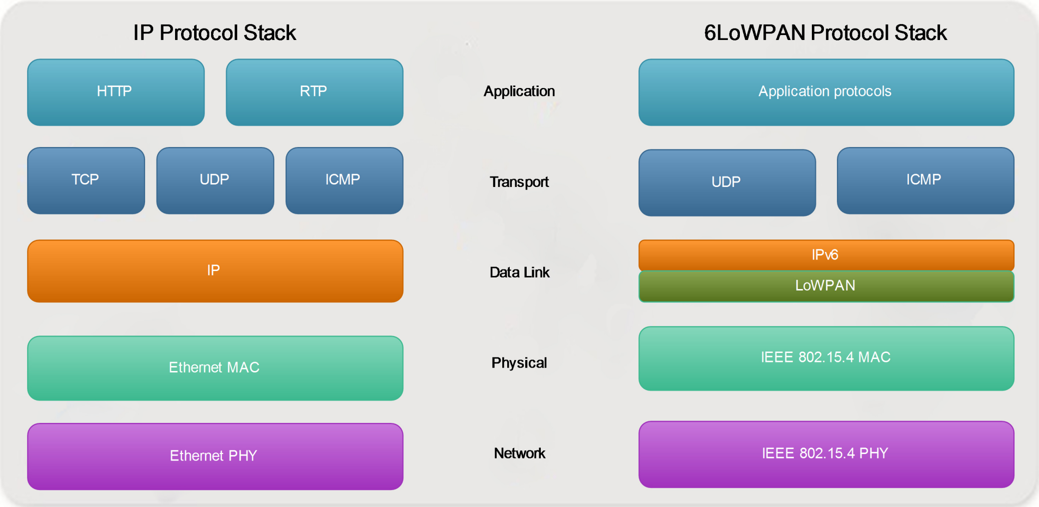


Figure 2.4: *IP and 6LoWPAN protocol stacks*

A simple IPv6 protocol stack with 6LoWPAN (also called a 6LoWPAN protocol stack) is almost identical to a normal IP stack with the following differences. First of all 6LoWPAN only supports IPv6, for which a small adaptation layer (called the LoWPAN adaptation layer) has been defined to optimize IPv6 over IEEE 802.15.4 and similar link layers in RFC6282 [2]. In practice, 6LoWPAN stack implementations in embedded devices often implement the LoWPAN adaptation layer together with IPv6, thus they can alternatively be shown together as part of the network layer. The most common transport protocol used with 6LoWPAN is the User Datagram Protocol (UDP), which can also be compressed using the LoWPAN format. The TCP is not commonly used with 6LoWPAN for performance, efficiency and complexity reasons. The Internet Control Message Protocol (ICMPv6) is used for control messaging, for example ICMP echo, ICMP destination unreachable and Neighbor Discovery messages.

Application protocols are often application specific and in binary format, although more standard application protocols are becoming available. Adaptation between full IPv6 and the LoWPAN format is performed by routers at the edge of 6LoWPAN islands, referred to as edge routers. This transformation is transparent, efficient and stateless in both directions. LoWPAN adaptation in an edge router typically is performed as part of the 6LoWPAN network interface driver and is usually transparent to the IPv6 protocol stack itself. Figure 2.5 illustrates one realization of an edge router with 6LoWPAN support.

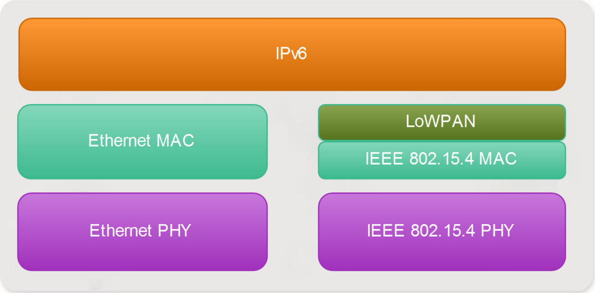


Figure 2.5: *IPv6 edge router with 6LoWPAN support.*

Inside the LoWPAN, hosts and routers do not actually need to work with full IPv6 or UDP header formats at any point as all compressed fields are implicitly known by each node [1].

Addressing. IP addressing with 6LoWPAN works just like in any IPv6 network, and is similar to addressing over Ethernet networks as defined by [RFC2464]. IPv6 addresses are typically formed automatically from the prefix of the LoWPAN and the link-layer address of the wireless interfaces. The difference in a LoWPAN is with the way low-power wireless technologies support link-layer addressing; a direct mapping between the link-layer address and the IPv6 address is used for achieving compression. This will be explained in Section 1.3.4.

Low-power wireless radio links typically make use of flat link-layer addressing for all devices, and support both unique long addresses (e.g. EUI-64) and configurable short addresses (usually 8-16 bits in length). The IEEE 802.15.4 standard, for example, supports unique EUI-64 addresses carried in all radio chips, along with configurable 16-bit short addresses. These networks by nature also support broadcast (address 0xFFFF in IEEE 802.15.4), but do not support native multicast.

IPv6 addresses are 128 bits in length, and (in the cases relevant here) consist of a 64-bit prefix part and a 64-bit Interface Identifier (IID) [41]. Stateless Address Autoconfiguration (SAA) [42] is used to form the IPv6 interface identifier from the link-layer address of the wireless interface as per RFC6775 [40]. For simplicity and compression, 6LoWPAN networks assume that the IID has a direct mapping to the link-layer address, therefore avoiding the need for address resolution. The IPv6 prefix is acquired through Neighbor Discovery Router Advertisement (RA) messages as on a normal IPv6 link. The construction of IPv6 addresses in 6LoWPAN from known prefix information and known link-layer addresses, is what allows a high header compression ratio.

Header format. 6LoWPAN compression is stateless, and thus very simple and reliable. It relies on shared information known by all nodes from their participation in that LoWPAN, and the hierarchical IPv6 address space, which allows IPv6 addresses to be elided completely most of the time. The LoWPAN header consists of a dispatch value identifying the type of header, followed by an IPv6 header compression byte indicating which fields are compressed, and then any in-line IPv6 fields. An example of 6LoWPAN compression is given in Figure 2.6.

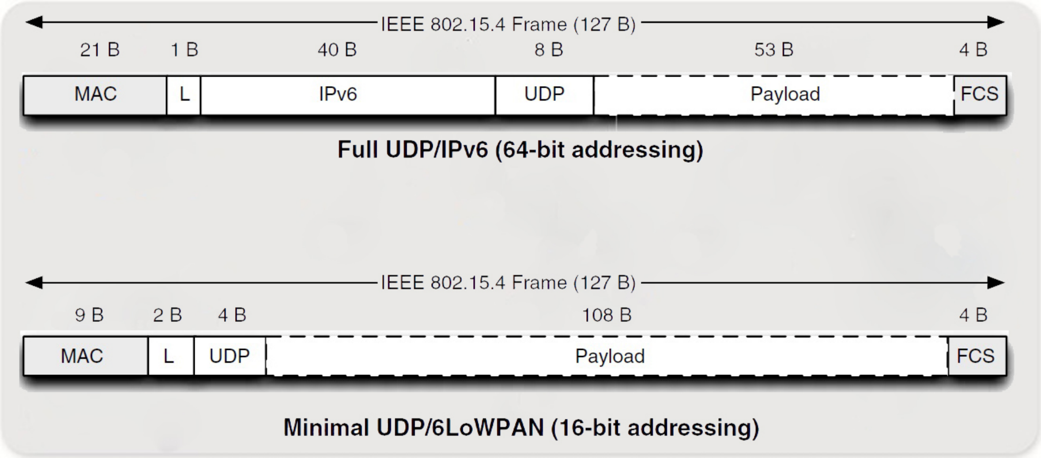

IEEE 802.15.4 Frame (127 B)
21 B
1 B
40 B
8 B
53 B
4 B
MAC
L
IPv6
UDP
Payload
FCS
Full UDP/IPv6 (64-bit addressing)
IEEE 802.15.4 Frame (127 B)
9 B
2 B
4 B
108 B
4 B
MAC
L
UDP
Payload
FCS
Minimal UDP/6LoWPAN (16-bit addressing)

Figure 2.6: *6LoWPAN header compression example (L = LoWPAN header) [1]*

In the upper packet a one-byte LoWPAN dispatch value is included to indicate full IPv6 over IEEE 802.15.4. Figure 2.7 gives an example of 6LoWPAN/UDP in its simplest form (equivalent to the lower packet in Figure 2.6) with a dispatch value and IPv6 header compression (LOWPAN_IPHC). The LOWPAN_IPHC encoding utilizes 13 bits, 5 of which are taken from the rightmost bits of the dispatch type. The encoding may be extended by another octet to support additional contexts. Any information from the uncompressed IPv6 header fields carried in-line follow the LOWPAN_IPHC encoding, as shown in Figure 2.7.

```
+----------+-------------+------------+--------------------+
| Dispatch + LOWPAN_IPHC (2-3 octets) | IPv6 Header Fields |
+----------+-------------+---------------------------------+
```

Figure 2.7: *LOWPAN_IPHC Header [2]*

In the best case, the LOWPAN_IPHC can compress the IPv6 header down to two octets (the dispatch octet and the LOWPAN_IPHC encoding) with link-local communication. When routing over multiple IP hops, LOWPAN_IPHC can compress the IPv6 header down to 7 octets (1-octet dispatch, 1-octet LOWPAN_IPHC, 1-octet Hop Limit, 2-octet Source Address, and 2-octet Destination Address). The Hop Limit may not be compressed because it needs to decremented at each hop and may take any value. Stateful address compression must be applied to the source and destination IPv6 addresses because they do not statelessly match the source and destination link-layer addresses on intermediate hops. [2] By comparison a standard IPv6/UDP header is 48 bytes in length as shown in Figure 2.6. Considering that in the worst case IEEE 802.15.4 has only 72 bytes of payload available after link-layer headers, compression is important.

6LoWPAN has been designed with IEEE 802.15.4 in mind. A well-targeted focus on that important link-layer technology was burned into the charter of the 6LoWPAN Working Group and has certainly helped the WG not to wander off into complex, hard to implement generalizations. The support for IEEE 802.15.4 can be considered to be a lead-in to a wider set of emerging standards: just as Ethernet has shaped other technologies in the link-layer space such as the IEEE 802.11 WLAN standards, there is good reason to expect that new specifications in the wireless embedded space will attempt to stay on a par with the feature set of IEEE 802.15.4, making 6LoWPAN applicable to a much wider set of technologies [1].

In wireless networks, the communication medium does not follow the binary characteristic of its wired counterpart where changes are rare. Instead, signal strength might vary due to energy levels, changes in the environment might interfere with a node's signal, or node mobility might cause changes in the network topology. As a result, a node's neighborhood in wireless ad-hoc networks might be constantly changing, causing communication to be time-variant.

A node in a Low Power and Lossy Networks (LLN) not only forwards its own packet towards the destination but also routes the packets of the other nodes in the network, routing is of great concern when considering preserving resources in these devices. A LLN contains several alternative paths towards a single destination, hence it becomes imperative of the routing protocol to make intelligent decisions while establishing the routes from a source to a destination. The poor path selection causes the scarce resources to drain out quickly. In the next chapter, a detailed description is made concerning the routing protocol used in this project.

2.2.3 Contiki and Cooja

There are various COTS operating systems implemented for low-power wireless networks. Among all the existing operating systems TinyOS and Contiki are more interesting as they provide various functionalities. The Contiki operating system was initially designed for IP-based networks. It has more facilities and extensions for IP-based protocols with a simple C programming. Hence, implementing the IP-based RPL is more convenient within Contiki.

Contiki operating system. Contiki encompasses kernel, libraries, the program loader, and a set of processes [43]. It is used in networked embedded systems and smart objects. Contiki provides mechanisms that assist in programming the smart object applications. It provides libraries for memory allocation, linked list manipulation and communication abstractions. Contiki is developed with C programming and thus it is highly portable to different architectures like Texas Instruments MSP430 microcontroller.

Contiki is an event-driven operating system in which processes are implemented as event handlers that run to completion. A Contiki system is partitioned into two parts: the core and the loaded programs. The core consists of the Contiki kernel, the program loader, the language run-time, and a communication stack with device drivers for the communication hardware [43].

The Program loader loads the programs into the memory and it can either obtain it from a host using communication stack or can obtain from the attached storage device such as EEPROM. The Contiki operating system provides modules for different tasks. It provides the routing modules in a separate directory ”contiki/core/net/rpl” and consists of a number of files. These files are separated logically based on the functionalities they provide for instance rpl-dag.c contains the functionality for Directed Acyclic Graph (DAG) formation, rpl-icmp6.c provides functionality for packaging ICMP messages etc.

Cooja Simulator. Cooja is similar to TOSSIM in such a way that its main purpose is to simulate behavior of an operating system. Cooja is Java-based simulator developed for simulations of sensor nodes running operating system Contiki. Each node in the simulated network can be different not only concerning its installed software but also the hardware platform may vary. Cooja is a flexible simulator and many parts may be replaced or extended [44]. On the other hand, some crucial functions (e.g. radio models) are still waiting for the extension to the best knowledge of the author of this thesis. One of the differentiating features is that Cooja allows for simultaneous simulations at three different levels: Network Level, Operating System Level and Machine code instruction level [44]. Cooja can also run Contiki programs either compiled natively on the host CPU or compiled for MSP430 emulator.

The authors of Cooja claim that their simulator can work on different levels - that it enables the so-called cross level simulations [44]. For example ns-2 (networking level) is principally simulator designed for network and application levels without taking the hardware properties into its account while TOSSIM (operating system level) is intended particularly for simulating the behavior of the operating system TinyOS. Cooja provides simulations on all mentioned levels and the short description of them follows.

Networking level. This level is useful especially for developers of routing or other network protocols where specific behavior of the hardware is not such an important issue. Radio propagation and radio devices are the most important parts of this level. The users of Cooja may develop and exchange certain modules. The specific sensor nodes can be replaced by abstract Java implementations so that there is no connection with the operating system Contiki. Heterogeneous network consisting of the nodes running native code together with some nodes easily implemented in Java may be created [44].

Operating system level. The aim of this level is to simulate Contiki by executing native operating system code. This can be useful especially for the developers of Contiki to allow testing and evaluation of changes in Contiki libraries [44].

Machine code instruction set level. Nodes having different underlying structure may be simulated using Java-based microcontroller emulator instead of a compiled Contiki system. The emulator represents ESB (Embedded Sensor Board) node.

Cooja supports simulations at all these three described levels but each node can be simulated at only one of these levels. In one simulation, however, nodes can cooperate from all levels - i.e. an emulated node can send a radio packet to a Java based node [44].

2.3 Mobility in low-power wireless networks

The high amount of research and technological investments in WSNs has enabled many applications, ranging from monitoring environments in agricultural fields and buildings to event detection for fire/flood emergencies and target tracking in surveillance. A conventional WSN consists of a dense and large number of battery powered sensor nodes. The main task of these sensors is to (i) sample a physical quantity from the environment, (ii) process the data, and (iii) send the data through wireless communication to the destination node [45].

The traditional WSN architectures were based on the assumption of a dense network with static nodes. In this classic design, the static nodes can only communicate through a multi-hop to reach a destination. The recent research trends show mobility as an option for WSNs. In fact, these studies are mainly focused on the positive impact of mobile nodes for sensor networks to improve challenges, i.e., connectivity, cost, reliability and energy efficiency [28, 29, 30]. However, mobility can raise some other challenges in which the contact detection is on top of them. A guaranteed and good communication is possible in the existence of a good link quality between two nodes. When a node moves, detecting the best moment for transferring data in any direction (from the static to the mobile node or vice versa) is challenging. The current research plan addresses the problem of collecting data from mobile sources by access points, which is known as hand-off.

In this Section, we first introduce some application examples and show the need of data collection with mobile elements in the network. Then we describe the hand-off process as a technique to deliver data from the reading sensors to the fixed APs.

2.3.1 Application Examples

Given the age of many industrial manufacturing systems, intelligent and low-cost automation would improve the productivity and efficiency drastically. Traditional industrial automation systems are realized by their wired communication, which require expensive communication cabling with regular maintenance. The need of mobility in such environments is also a major challenge in installing such networks. Thus, the costly devices with expensive service system and implementation process reduces the automation tendency in industries. Therefore, there is a need to enable a wireless automation system that can handle mobility and obtain cost-effective industrial system [46].

In a commercial warehouse, sensor nodes can reduce the cost of operations remarkably. The deployed sensors can collect information for decision making. It is not uncommon that a forklift collides with warehouse walls, or other forklifts. The frequent collisions cause damages to the warehouse management. To detect the collision, the movement of forklifts can be monitored and alerted by embedded sensor nodes. A number of factors affect the dispatching process. The type of products that are supposed to be moved and the battery level of the forklift are two simple factors that can be notified by wireless sensors to enhance dispatching [47].

In military context, sensors are embedded on the body of soldiers and tanks. The readings by these end-points are forwarded to a fixed infrastructure of static nodes that are previously deployed in the battlefield. The location of these moving objects are randomly changing and their is no need to track the position of each node in order to receive data. Obviously, the sink node is accessible via fixed node through a multi-hop data collection [48].

In cities with limited parking lots, there is a high requirement of open-space smart parking platform. In order to establish such network, static sensor nodes at parking spaces should collect information to be delivered to the motorists with equipped wireless sensors. The users with mobile element can collect data from the fixed access points attached to the street lights [49].

Clinical deterioration of patients is a major concern in hospitals. Most of these patients need continuous monitoring by collecting events such as cardiac and respiratory arrests with high data rate, temperature , blood pressure and pulse with low data rate [50, 51, 52]. An early and retrospective detection of clinical deterioration prevents nearly 70% of harmful damages [53]. The detection is possible only by monitoring patients in Intensive Care Units (ICUs) to collect and study the vital signs. The scarcity of these wired and costly devices in ICUs prevents monitoring all patients with risky situation. The conventional method is to measure manually at long-term intervals, which is not a safe solution. A naive idea is to develop a Wi-Fi system with a wired infrastucure. The cost of deploying a mesh network with wireless sensors is much lower than a Wi-Fi network [9]. However, there is a challenging issue in reliably delivering the monitored data to the fixed sensor nodes.

2.3.2 Related Works on Mobility Management

Networks with mobility support require a mobility management mechanism in order to handle the sudden changes. In this work, we tackle the hand-off process that enables the one-hop data delivery from the source node to the best access point. Hand-off mechanism has been widely studied in cellular networks [15, 16, 17, 18, 19] and wireless local area networks [20, 21, 54, 22], but it has not received the same level of attention in WSNs.

In cellular networks, the hand-off decision is centralized and typically coordinated by a powerful base-station, which is able to leverage considerable information about the network topology and client proximity [15]. Cellular networks also take advantage of sophisticated CDMA radios to perform soft hand-off techniques [16]. The major challenge in cellular networks with hand-off support is the *call dropping* effect during an ongoing call while switching between base-stations [17]. A similar event occurs due to the lack of available channel –so-called *call blocking*. In [19], some channels are exclusively allocated to hand-off calls, also known as *guard channels*. In [18], a queuing strategy has been applied to delay the hand-off calls until a channel becomes available. Contrary to these resourceful systems, WSNs have constrained energy resources and simple single-channel radios, which require different solutions.

Contrary to cellular systems, WiFi networks have a distributed architecture, where mobile nodes have no a-priori knowledge of the local network [20, 21]. While cellular systems require a continuous monitoring of the signal level, WiFi-based systems monitor the signals only after service degradation. The main concern of 802.11 hand-off protocols is to minimize the hand-off latency for real-time applications. A hand-off process in WiFi-based systems is divided into the Discovery and Re-authentication phases. The channel scanning during a Discovery Phase is the most time consuming process. The authors in [54] propose a MAC layer with fast hand-off that uses selective scanning and records the scan results in AP's cache. When a MN moves to a location visited before, it pings the nearby APs for their available channels. In [22], each AP records the neighboring AP's information in a *neighbor graph* data structure. Then the AP can inform MN about which channels have neighboring APs. The MN needs to scan only those channels.

The key difference between WiFi and WSN hand-offs is that in WiFi multiple radios are used to reduce the hand-off latency while in WSN applications a single radio is used. In WSN, a centralized hand-off approach is not feasible as it incurs a high overhead on the system. hand-offs in sensor networks should be distributed –similar to WiFi networks– while using a single-channel radio that focuses on the up-link and that can cope with the high variability of low-power links.

There are two major strategies to make a hand-off process that are soft hand-off with network layer solution and hard hand-off with MAC layer solution. The first approach that neglects the energy conservation consideration has been extended in [55, 56].

In [55] the problem related to the mobility of sensor node (SN) to hand-off between different gateways (GW), connected to the backbone network is addressed. It proposes a soft hand-off decision for WSNs based on 6LoWPAN (SH-WSN6), which avoids unnecessary hand-offs when there are multiple GWs in the range of SNs. The sensor node is able to register to multiple GWs at the same time by using IP solution. The SH-WSN6 takes advantage of router advertisement (RA) message defined in the Internet Control Message Protocol (ICMP). GWs transmit RA messages periodically to advertise their presence. At first, SN can register to only one GW. By receiving RA in each interval, the SN decides for the best GW. Every time an SN registers with a new GW, it gains a new route. This improves connectivity by having route diversity. If there is an unreliable link, comparison algorithm makes a decision to remove that link and therefore improves the QoS since poor links will not be used anymore. Comparison algorithm makes independent decision for start of hand-off. Decision is made based on the comparison of the ratio of RA messages coming from GWs in the range. SN also notices when a GW moves away from SN's range by comparing the ratio of RA messages. Comparison algorithm assumes that GW's send RA messages at the same rate, which is a reasonable assumption.

In [56] two additional control messages are transmitted in order to support the attachment of the MN to a new point of attachment. These messages are the Join and the Join Ack that are sent/received when the MN is still attached to the previous tree position. Therefore, the role of the dynamic topology control in soft-hand-off mobility is to support the re-attachment of the MN to a different tree position as a result of movement inside the test-bed area. In the hand-off decision rules some parameters are defined which are (i) RSSI threshold, (ii) better RSSI, (iii) number of lost packets, and (iv) packet loss percentage. These values are set according to the application requirements.

The second approach that is more reasonable, addresses a MAC layer solution for hard hand-off mechanism in mobile WSNs. These solutions are either specialized for passive decision with non-real-time support in [9] or for active decision with real-time support in [10].

In [9] authors describe a wireless clinical monitoring system collecting the vital signs of patients. In this study, the mobile node connects to a fixed AP by listening to beacons periodically broadcasted by all APs. The node connects to the AP with the highest RSSI. The scheme is simple and reliable for low traffic data rates. However, there is a high utilization of bandwidth due to periodic broadcasts and hand-offs are passively performed whenever the mobile node cannot deliver data packets.

A reliable hand-off depends significantly on the link quality estimator used to monitor the link. Different link quality estimators have been proposed for sensor networks. They apply different criteria to estimate the link status, such as RSSI, SNR, LQI or link asymmetry [57, 11]. In our case we use a simple and fast sampling of RSSI and SNR, which have been shown to provide reliable metrics [12, 23].

In the next section, we explain the smart-hop design and implementation. Different phases of the hand-off design together with the parameters tackled are described extensively.

3

Basics on RPL

Routing algorithms are used to determine the paths the data will take and should fulfill the following properties: the routes should be chosen such that data reaches its destination in the 'best' way possible. 'Best' is defined by one or more metrics, depending on the application requirements. For example, one widely used metric is using the route with the lowest end-to-end delay, or the highest throughput, whilst other ones could be to use the route with the least hop distance, the best link quality, or least energy consumption. RPL has been proposed by the IETF ROLL Working Group as a standard routing protocol for IPv6 routing in low-power wireless sensor networks.

Existing routing protocols such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), Ad hoc On-Demand Distance Vector Routing (AODV), and Optimized Link State Routing Protocol (OLSR) have been extensively evaluated by the working group and have been found to not satisfy, in their current form, all specific routing requirements for LLN [58]. Finding the best routes for the delivery of data implies a very efficient routing mechanism for determining and keeping the routes in the network. Routing is then a key feature in a LLN, and since this is the main object of this work, a special emphasis is done in this chapter.

3.1 Overview

RPL is an IPv6 based distance vector routing protocol for low-power and lossy networks (LLNs) [4]. RPL is a distance vector protocol and unlike linked state protocols does not require significant amount of memory, which is not suitable for resource constrained LLNs. RPL is a proactive routing protocols and starts finding the routes as soon as the RPL network is initialized. RPL forms a tree like topology also called DAG. The DAG defines a tree-like structure that specifies the default routes between nodes in the LLN. However, a DAG structure is more than a typical tree in the sense that a node might associate to multiple parent nodes in the DAG, in contrast to classical trees where only one parent is allowed.

More specifically, RPL organizes nodes as Destination Oriented Directed Acyclic Graph (DODAG), where most popular destination nodes (i.e. sinks) or those providing a default route to the Internet (i.e. gateways) act as the roots of the DAGs. A network may consist of one or several DODAGs, which form together an RPL instance identified by a unique ID, called *RPLInstanceID*. A network may run multiple RPL instances concurrently; but these instances are logically independent. A node may join multiple RPL instances, but must only belong to one DODAG within each instance [3]. Each such instance may serve different and potentially antagonistic constraints or performance criteria.

Each node in an RPL network has a preferred parent which acts like a gateway for that node. If a node does not have an entry in its routing table for a packet, the node simply forwards it to its preferred parent and so on until it either reaches the destination or a common parent which forwards it down the tree towards the destination. The nodes in an RPL network have routes for all the nodes down the tree. It means the nodes nearer to the root node have larger routing tables. Route aggregation is not recommended because of several problems in LLN like mobility of nodes and losses in the radio medium. Path selection is an important factor for RPL and unlike traditional networks routing protocols, RPL uses more factors while computing best paths for example routing metrics, objective functions and routing constraints.

3.2 Routing metrics

A metric is a scalar quantity used as input for best path selection. A constraint, on the other hand, is used as an additional criterion to prune links or nodes that do not meet the set of constraints. Commonly used metrics for routing are hop count, energy, Expected Transmission Time (ETT), and Expected Transmission Count (ETX).

Minimum Hop Count is a most common metric used in routing protocols, where routing protocol find the path from sender to receiver that have minimum number of hops (shortest hop). In this route selection metric, routing protocol never consider the link cost and select the path that involved the smallest number of forwarding nodes and minimizes the total data propagation cost from sender to receiver. Minimum-hop based routing protocol provide no optimal route in terms of congestion, delay, and energy because it never consider the resource availability on each node.

Energy is the most critical resource available in LLNs which makes it more challenging to propagate a single packet from source to a destination by using minimum energy. Sometimes a node with minimum available energy is avoided to select as a router which may result in non optimal or longer paths. To maximize the lifetime of a whole network depends upon the equally distribution of energy on all network nodes.

ETT is an estimation of time cost of sending a packet successfully through a MAC layer. It takes the link bandwidth into account and commonly used to express latency.

ETX is defined as the expected number of MAC layer transmissions necessary to successfully delivering a packet through a wireless link. If the link quality/Packet Delivery Ratio (PDR) is high, the expected number of transmissions to reach the next hop may be as low as 1. However, if the PDR for the particular link is low, multiple transmissions may be needed [59].

Routing for an LLN requires a sophisticated routing metric strategy driven by type of data traffic. The metrics and constraints can be dynamic and the routing protocol "smoothes" and reacts to the changes in metric and constraint values.

3.3 RPL messages

RPL build whole topology graph by using three different ICMPv6 based control messages: the DODAG Information Object (DIO), the DODAG Destination Advertisement Object (DAO), and the DODAG Information Solicitation (DIS). RPL message structure is depicted in 3.1.

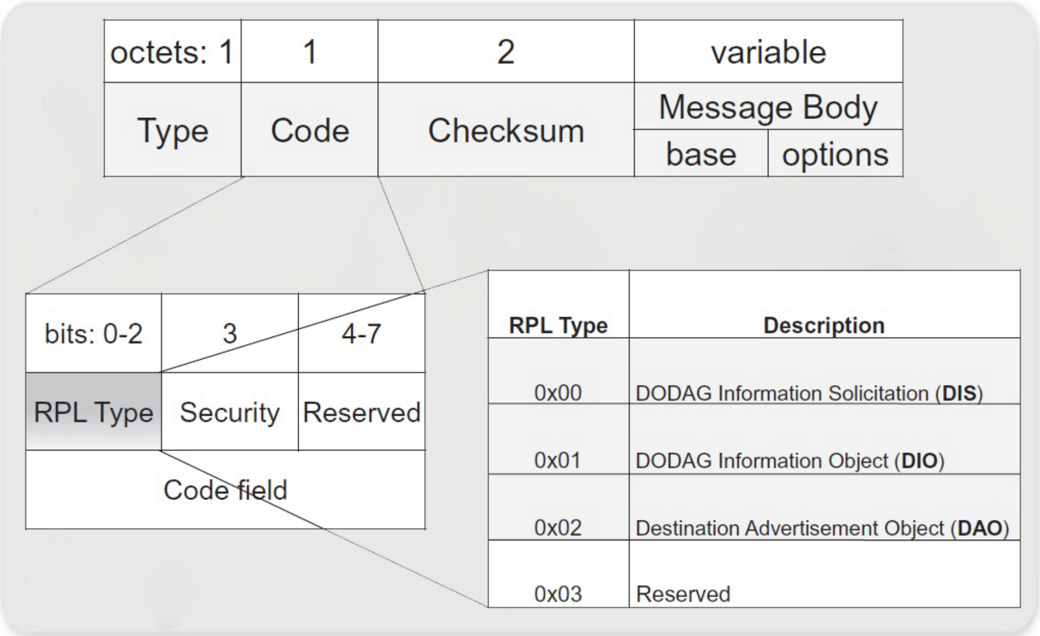


Figure 3.1: *RPL control message [3]*

The RPL control message is composed of (i) an ICMPv6 header, which consists of three fields: Type, Code and Checksum, (ii) a message body comprising a message base and a number of options. The Type field specifies the type of the ICMPv6 control message prospectively set to 155 in case of RPL [4]. The Code field identifies the type of RPL control message. Four codes are currently defined:

- DIS/ The DIS message is mapped to 0x00, and is used to solicit a DIO from an RPL node. The DIS may be used to probe neighbor nodes in adjacent DODAGs. The current DIS message format contains non-specified flags and fields for future use as depicted in 3.2.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |    Reserved   |   Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.2: *The DIS Base Object [4]*

This type of message is usually used when nodes join a network. As an alternative to waiting to receive a DIO message, a node can choose to broadcast a DIS message so that other nodes will immediately trigger a DIO transmission upon receiving the DIS message.

- DIO. DIO messages are sent as link-local multicast toward other neighboring nodes in downward direction and enable Point-to-Multipoint traffic in upward direction. DIO messages contain the root nodes identity, routing metrics, rank, objective function and DODAG-ID. These messages are sent periodically with increasing sequence number in order to start the parent selection process. The format of the DIO Base Object is presented in 3.3.

```
 0               1               2               3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |Version Number |             Rank              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|0| MOP | Prf |     DTSN      |     Flags     |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID                            +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 3.3: *The DIO Base Object [4]*

The main DIO Base Object fields are: (i) RPLInstanceID, is an 8-bit information initiated by the DODAG root that indicates the ID of the RPL instance that the DODAG is part of, (ii) Version Number, indicates the version number of a DODAG that is typically incremented upon each network information update, and helps maintaining all nodes synchronized with new updates, (iii) Rank, a 16-bit field that specifies the rank of the node sending the DIO message, (vi) Destination Advertisement Trigger Sequence Number (DTSN) is an 8-bit flag that is used to maintain downward routes, (v) Grounded (G) is a flag indicating whether the current DODAG satisfies the application-defined objective, (vi) MOP identifies the mode of operation of the RPL instance set by the DODAG root. Four operation modes have been defined (3.4) and differ in terms of whether they support downward routes maintenance and multicast or not. Any node joining the DODAG must be able to cope with the MOP to participate as a router, otherwise it will be admitted as a leaf node.

A value of 0 indicates that destination advertisement messages are disabled and

```
+-----+-----------------------------------------------------+
| MOP | Description                                         |
+-----+-----------------------------------------------------+
|  0  | No Downward routes maintained by RPL                |
|  1  | Non-Storing Mode of Operation                       |
|  2  | Storing Mode of Operation with no multicast support |
|  3  | Storing Mode of Operation with multicast support    |
|     |                                                     |
|     | All other values are unassigned                     |
+-----+-----------------------------------------------------+
```

Figure 3.4: *MOP Encoding [4]*

the DODAG maintains only Upward routes. (vii) DODAGPreference (Prf) is a 3-bit field that specifies the preference degree of the current DODAG root as compared to other DODAG roots. It ranges from 0 ” 00 (default value) for the least preferred degree, to 0 ” 07 for the most preferred degree, (viii) DODAGID is a 128-bit IPv6 address set by a DODAG root, which uniquely identifies a DODAG. Finally, DIO Base Object may also contain an Option field.

- DAO. The DAO message is used to propagate reverse route information to record the nodes visited along the upward path. DAO messages are sent by each node, other than the DODAG root, to populate the routing tables with prefixes of their children and to advertise their addresses and prefixes to their parents. After passing this DAO message through the path from a particular node to the DODAG root through the default DAG routes, a complete path between the DODAG root and the node is established. 3.5 illustrates the format of the DAO Base Object.

  As shown in the figure, the main DAO message fields are: (i) RPLInstanceID, is an 8-bit information indicates the ID of the RPL instance as learned from the DIO, (ii) K flag that indicates whether and acknowledgment is required or not in response to a DAO message, (iii) DAOSequence is a sequence number incremented at each DAO message, (iv) DODAGID is a 128-bit field set by a DODAG root which identifies a DODAG. This field is present only when flag D is set to 1.

- DODAG Destination Advertisement Object Acknowledge (DAO-ACK). The DAO-ACK message is sent as a unicast packet by a DAO recipient (a DAO parent or DODAG root) in response to a unicast DAO message. It carries information about RPLInstanceID, DAOSequence, and Status, which indicate

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |K|D|   Flags   |   Reserved    | DAOSequence   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID*                           +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 3.5: *The DAO Base Object [4]*

the completion. Status code are still not clearly defined, but codes greater than 128 mean a rejection and that a node should select an alternate parent [4, 3].

3.4 Objective function

An Objective Function (OF) defines how routing metrics, optimization objectives, and related functions are used to compute Rank. Furthermore, the OF dictates how parents in the DODAG are selected and, thus, the DODAG formation. Objective Function is used in RPL to construct the DODAG and define how nodes in RPL select the routes within an instance. RPL define the whole topology by constructing DODAGs with instances. Each instance is associated with a specialized objective function.

Objective Function combines the metrics and constraints to find the best path. Consider a physical network made of several links with different qualities such as throughput, Latency and nodes with different qualities such as battery-operated, mains-powered. If the network carries different types of traffic, it might be useful to carry the traffic based on different OFs, which are optimizing different metrics or fulfilling constraints. For example, the objective function finds the path that has minimum delay and that path never traverse a battery-operated node. In this example, path with minimum delay is representing the metric and non-battery operated nodes are representing the constraint.

Each Objective Function is identified by Objective Code Point (OCP) in a DIO configuration option. Objective function is also used to define the rank of a node which is a nodes distance from a DODAG root node. RPL implements two OFs (OF0 and ETX). OF0 uses hop count as routing metric. This separation of OFs from the core protocol specification allows RPL to be adopted to meet the different optimization criteria required for a wide range of deployments, applications and network designs. Objective Function 0 (OF0) is designed as a default function that is common to all implementations and provide interoperability between different implementations [4, 60, 61].

3.5 Topology

Topology formation in RPL starts with designating one node as a root node. The root node determines the configuration parameters for the network. The configuration is packed into a DIO message, which is then used to disseminate the information in the network. There are many options that can be configured in a DIO to tailor the network configuration to the application's requirements. The root node triggers the DODAG formation by broadcasting a DIO message to its neighbors 3.6. Note that only the root node of a DODAG is allowed to initiate the diffusion of DIOs.

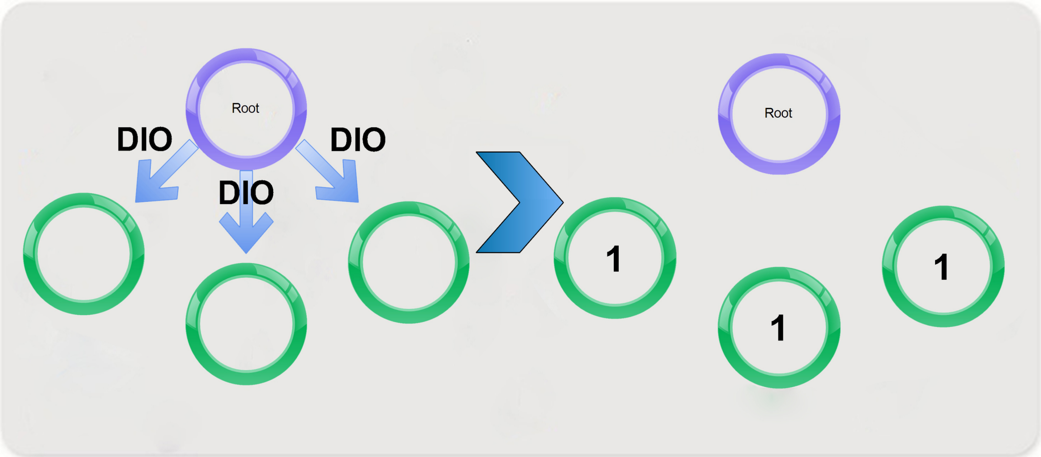


Figure 3.6: *DIO message is broadcast by the root node to its neighbors. After receiving a DIO from the root, each neighbor calculates its rank by computing its hop count distance to the root node (in this case, rank equals to 1)*

Whilst the RPLInstanceID and the DODAGID remain unchanged throughout the whole topology formation, the rank field is updated, as the DIO messages are traversing the network. Since the root node has a distance of 0 to itself, its rank is set to 0. Each neighbor receiving the DIO, calculates its rank according to the OF by computing its hop count distance to the root node and sets its rank to 1. After calculating its rank, each node updates the DIO and broadcasts it to its neighbors 3.7. Each node retains a candidate neighbor set, in which it keeps track of the neighbors with lower or equal rank from whom they received a DIO message. Out of this candidate neighbor set, each node selects parent nodes, which have to have a lower rank than the node itself.

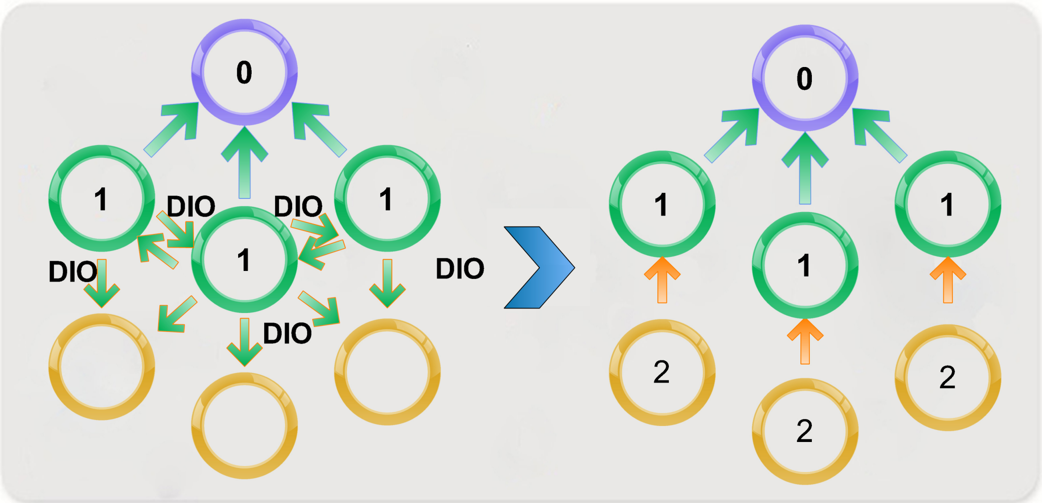


Figure 3.7: .
Rank 1 nodes broadcast a DIO to their neighbors, originating rank 2 nodes and completing the network topology

From the parent set, the node picks a so-called preferred parent, which serves as the node's next hop when routing a data packet towards the root. This choice is determined by the OF. In the example the neighbors of the root node only know of one node fulfilling this condition, so they pick the root as their preferred parent. Nodes with rank 2, choose the neighbors with rank 1 as their parents.

With all nodes having joined the DODAG, the topology formation is complete - for this iteration, which was initiated by the root node. It can happen that node failures or changing environmental conditions create the need to rebuild the routing topology. To help the nodes keep track of which DODAG iteration they are in, and to determine whether it is the newest one, a version number is written in the DIO message. Note that only the root node is allowed to increment the version number in order to trigger a rebuild of the DODAG. So whenever a node receives a DIO message containing a newer version number than the one it recorded, it can add the sender of this DIO to its candidate neighbor set and might even select it as parent.

However, a node can only become part of the new DODAG iteration - and advertise the appropriate version number - once all its parents are part of the new iteration as well. This switch again is governed by the OF which could for example define that a certain percentage of a node's parents need to be a member of the new DODAG iteration before the node is allowed to switch and to discard all of its outdated parents. However, the mere fact that a node detects a clash in version numbers indicates changes in the network which have to be consolidated by ensuring that all nodes are updated to the current DODAG version. This task is crucial, yet challenging in that it requires reliability when disseminating new information whilst still aiming to be performed at a minimal overhead on the protocol. For meeting these goals, RPL is employing the Trickle algorithm [62].

3.6 Trickle timer

The Trickle algorithm allows nodes in a lossy shared medium (e.g., low-power and lossy networks) to exchange information in a highly robust, energy efficient, simple, and scalable manner. Dynamically adjusting transmission windows allows Trickle to spread new information on the scale of link-layer transmission times while sending only a few messages per hour when information does not change. A simple suppression mechanism and transmission point selection allow Trickle's communication rate to scale logarithmically with density [63, 64].

Trickle Timer is used to control the sending of DIO and DAO messages. It is based on dynamic timers that govern the transmission of RPL control messages in energy effcient, and scalable manner and also to reduce redundant messages. Trickle timer control the inconsistency and avoid redundant transmissions of DIO messages. In case of instability in DODAG, trickle time interval become shorter and control messages are sent frequently to stabilize the DODAG. Similarly, when DODAG becomes stable, control messages are less frequent to reduce the control plane overhead [60].

Reducing transmissions in dense networks conserves system energy. To save energy the DIOs are sent periodically controlled by the trickle timer whose duration is doubled each time it is fired. The smallest possible interval between two DIOs equals to DIO Minimum Interval which keeps on increasing (doubling) until it reaches the maximum value determined by DIO Interval Doublings. There are three configurable parameters in the Trickle Timer.

- Imin: This parameter gives the minimum amount of time between two DIOs. DIOs are transmitted periodically to reduce the redundant control traffic and use the limited resources more efficiently. The value of trickle timer starts from the lowest possible value Imin and is doubled each time it is transmitted until it reaches its maximum possible value of Imax. The value of Imin is determined by the RPL parameter DIO Minimum Interval and computed as: $Imin = 2^{RPL\_DIO\_INTERVAL\_MIN}$. So if $RPL\_DIO\_INTERVAL\_MIN = 12$ then $Imin = 2^{12} = 4096ms = 4s$.

- Imax: This parameter is used to limit the number of times the Imin can be doubled. So if $RPL\_DIO\_INTERVAL\_DOUBLINGS = 8$ and Imin is 4096 then $Imax = 4096 * 2^8 = 1048576ms = 17.5min$. This is the maximum time between two successive DIOs required under a steady network condition.

- Redundancy constant (k): It is a natural number greater than 0 and is used to suppress the DIO transmission. In RPL, when k has the value of 0x00, this is to be treated as a redundancy constant of infinity in RPL, i.e., Trickle never suppresses messages [4, 61].

Core technologies and parameters have been described with adequate detail to provide a good comprehension of the following chapters. Next, we present the mobility subject and its relevance to real-life applications.

4

smart-HOP

4.1 smart-HOP Algorithm

The gist of this algorithm is to devise a reliable data collection process from the mobile nodes in a wireless sensor network. The reliability is defined in terms of delivering packets in a timely basis from the mobile node to the destination. smart-HOP provides these features by means of employing hand-off process.

In this section, first we provide the overall idea of smart-HOP and highlight the importance of three parameters: link monitoring, hysteresis thresholds and stability monitoring, an then, we describe the tuned parameters in a controlled environment. The smart-HOP algorithm has two main phases: *(i) Data Transmission Phase* and *(ii) Discovery Phase.* A timeline of the algorithm is depicted in Figure. 4.1.

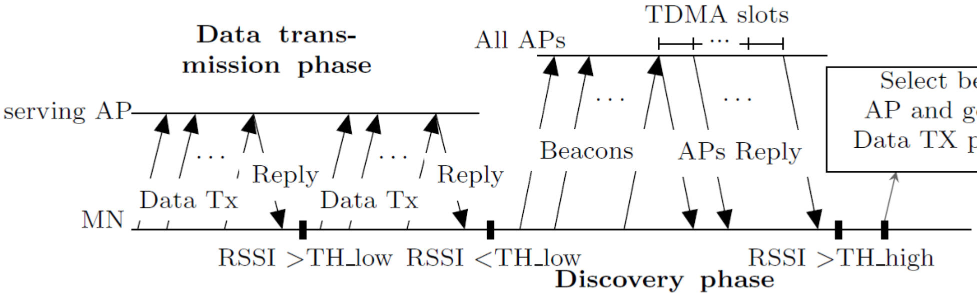


Figure 4.1: *Time diagram of the smart-HOP mechanism*

Initially, the mobile node is not attached to any access point. This state is similar to the case when the MN disconnects from one AP and searches for a better AP. In both cases, the MN performs a Discovery Phase by sending n request packets in a given window w and receiving a *reply* packet from each neighboring AP. The reply packet holds the link quality level that is defined as the average received signal strength (RSSI), or signal-to-noise ratio (SNR), of the n packets. By embedding the link quality level to the reply packet, the MN gets the down-link information and filters out the asymmetric links. Upon detecting a good link, the MN resumes a Data Transmission Phase with the AP serving the most reliable link. The data packets are sent in burst and receive a reply afterwards similar to the Discovery Phase. This process enables monitoring the current link during the normal data communication process. The details of both phases are shown in Figure 4.1. The smart-HOP process relies on three main tuning parameters, which are presented in details as follows.

Parameter 1: link monitoring frequency. It is an important parameter for any hand-off process, which determines how frequent the link monitoring should be. The link monitoring property is captured by the Window Size parameter (ws), which represents the number of packets required to estimate the link quality over a specific time. Considering an inter-packet interval of 10 ms with ws=3, results in link monitoring frequency of 33 Hz. A small ws (high sampling frequency) provides detailed information about the link but increases the processing of reply packets, which leads to higher energy consumption and lower delivery rates. The packet delivery reduces as the MN opts for several unnecessary hand-offs. The hand-off is ordered by detecting low quality links that happens by sudden fluctuations of signal strength. On the other hand, a large ws (low sampling frequency) provides only coarse grained information about the link and decreases the responsiveness of the system. A large ws leads to late decision, which is not suitable for a mobile network with dynamic link changes.

The mobile node starts the Discovery Phase when the link quality goes below a certain threshold (TH_{low}) and looks for APs that are above a reliable threshold ($TH_{high} = TH_{low}+HM$, where HM is the hysteresis margin). During the Discovery Phase, the mobile node sends ws beacons periodically each 10 ms (the minimum possible inter-packet interval), and the neighboring APs reply with the average RSSI or SNR of the beacons. If one or more APs are above TH_{high}, the mobile node connects to the AP with the highest link quality and resumes data communication, else, it continues broadcasting beacons in burst until discovering a suitable AP. In order to reduce the effects of collisions, the APs use a simple TDMA MAC. Our studies enclose that the smart-HOP running a TDMA-based MAC reduces hand-off delay compared with a CSMA-based MAC due to the collision avoidance feature of a TDMA-based scheme [10].

Parameter 2: threshold levels and hysteresis margin. In WSNs, the selection of thresholds and hysteresis margins is dictated by the characteristics of the transitional region and the variability of the wireless link. The lowest threshold has to consider the boundaries of the transitional region. Wireless sensors spend most of the time in the transitional region. The exact threshold level within the transitional region is computed from the simulation and experimental analysis. If threshold TH_{low} is too high, the node could perform unnecessary hand-offs (by being too selective). If the threshold is too low the node may use unreliable links. The hysteresis margin plays a central role in coping with the variability of low-power wireless links. If the hysteresis margin is too narrow, the mobile node may end up performing unnecessary and frequent hand-offs between two APs (ping-pong effect), as illustrated in Figure 4.1. If the hysteresis margin is too large, the hand-off may take too long, which ends up increasing the network inaccessibility time, and thus delivery delay and decreasing the delivery rate.

Parameter 3: AP stability monitoring. Due to the high variability of wireless links, the mobile node may detect an AP that is momentarily above TH_{high}, but the link quality may decrease shortly after being selected. In order to avoid this, it is important to assess the stability of the AP candidate. After detecting an AP above TH_{high}, smart-HOP sends m further bursts of beacons to validate the stability of that AP. The burst of beacons stands for the ws request beacons followed by the reply packets received from neighboring APs. Stability monitoring is tightly coupled to the hysteresis margin. A wide hysteresis margin requires a lower m, and vice versa.

Architectural design. smart-HOP has some distinct design features. Most hand-off methods perform explicit disconnections, i.e., the node informs the old AP that it no longer needs it. smart-HOP does not perform these disconnections for two reasons. First, sensor network deployments may have a limited overlap between neighboring APs – due to low coverage radios and low node density–, and this limited overlap may not permit complex transactions (by the time a mobile node wants to disconnect, the AP may already be out of range). Second, removing explicit disconnections reduces the computational and transmission costs of mobile nodes. Applications similar to cellular networks perform explicit disconnections because they provide circuit switching services (dedicated communication channel). We argue that for several applications envisioned in mobile sensor networks (reliable transfer of information from mobile nodes to a fixed infrastructure), hand-offs do not require explicit disconnections.

The lack of explicit disconnections implies that the fixed infrastructure is not responsible to track the connectivity of mobile nodes (as opposed to what happens in cellular networks). Hence, the mobile node should take an active role in avoiding disconnections. This is simply done by maintaining a disconnection time-out. If the mobile node does not receive *reply* packets for a certain period of time, it starts the discovery phase. The time-out parameter depends on the real-time requirements of the application, was set to 100 ms.

4.2 Implementation

As previously stated, smart-HOP runs a simple TDMA-based MAC on the APs in order to avoid collisions and reduce the hand-off delay. Each AP performs a simple modulo operation on its unique *id* to obtain a specific time-slot. The *id* of APs is in the range of $[1, 9] \times n$ where $n \in N$. By performing a modulo operation with value 10, the *id* of APs becomes in the range of $[1, 9]$, which takes at most 9 slots. The MN is assigned an *id* which is a multiple of 10, e.g. 0, 10, 20 and etc. In theory, two nodes could collide, for example APs with *id*s 14 and 24 would select the same time slot 4, in practice, clock drifts and a relatively low density of access points ($\leq$10) makes this unlikely.

The MAC scheme is not in the scope of our work. The main idea on choosing a simple TDMA-based MAC is to ignore packet collision. In smart-HOP evaluation, the modulo operator is 10; while in the preliminary experiments the length of time-slot was considered 5 ms and in the extended experiments it is raised to 10 ms. The preliminary experiments were performed in a controlled environment with toy train, which enables repeatability of the experiment. The extended experiments is conducted in a realistic environment, which a person holds the mobile node walking in a room. The time-slot increase is mainly due to the additional processing time needed to guarantee the following features: *(i) prevent failure cases which will be described later, (ii) support many APs, and (iii) support many MNs.*

smart-HOP was implemented in TinyOS 2.0.2 [65] and used telosB [66] motes for the evaluation. In TinyOS, packet-level communication has two main classes of interfaces; send and receive. Each type of device (MN and AP) should handle these events according to their specific job in smart-HOP implementation. For instance, the MN is supposed to broadcast beacons in the Discovery Phase and then switch to unicast transmission of data in the Data Transmission Phase. On the other hand, the APs should process the packets that are received by MNs and then *reply* packets should be directed to the requested MN only with customized information. Existence of different phases on each device increases the complexity of run-time computations and processing. The nodes should be smart enough to process the computation and communication with the limited memory and processing capability.

Initially, there is not any association between different devices. For instance, the MN is an orphan node which is not attached to any AP. After the first deployment of nodes, the MN starts broadcasting the beacons in the Discovery Phase which is implemented as $BeaconTimer$ event in TinyOS —it appears in Algorithm (1) in the appendix. The access of device to each phase is handled by applying two flags; namely $DiscoveryPhase$ and $DataPhase$, which are set to TRUE and FALSE respectively when the radio starts. This forces the device to enter the $BeaconTimer$ after calling the Boot event. In the Discovery Phase of the enhanced smart-HOP algorithm, w beacons are sent in burst with the time interval of 10 ms, but the timer expires 100 ms after sending the last beacon. The intuition is to enable receiving at most 9 reply packets from the neighboring APs (modulo operator=10) within this period. However, in the Data Transmission Phase, only 10 ms is enough to receive the reply packet from the corresponding AP.

The *DataTimer* ,Algorithm (2) in the Appendix, establishes a unicast communication with the selected AP by sending data to the node with stored id. The MN keeps assessing the current link within each ws. The RSSI value is retrieved from the received packet and compared to TH_{low}. If the current AP is not qualified for the rest of communication, the *DataTimer* is stopped, the starting moment of hand-off is stored, numbers of hand-offs increases by one, flags are changed and the $BeaconTimer$ resumes.

At the AP side, the main tasks are performed at the receive event as smart-HOP assumes that they are activated when receiving a packet from a MN —see Algorithm (3) in the Appendix. Each time after receiving a packet, some checking should be done.

- Is the received packet sent from a MN device?
- Does the packet sent during a Discovery Phase or Data Transmission Phase?
- Is the AP in the middle of a communication or it is just the beginning?
- Check the sequence number of the packet which changes the replying slot.

Each group of packets in a sliding window has same sequence number with counters from 1 to the number of ws. The sequence number is advanced in the next sliding window. smart-HOP considers some watchdog timers, which prevent possible failures during a communication. It may happen that the AP does not receive all the packets within a window size. Instead of waiting for all packets, it starts various timers after receiving each packet. Immediately after receiving a packet at the AP, a timer is called which is fired after a predefined amount of time. Before ending this waiting period, if the AP receives a packet, the timer stops and if it does not receive, a reply packet is sent back to the requested MN.

4.3 Related parameters

Some experiments were conducted with narrow and wide hysteresis margin. In each case, they considered different threshold levels for starting a hand-off; i.e. -95, -90, -85 and -80 dBm. The stability monitoring parameter was also examined by assuming m=1, 2, and 3. The results in [10] indicate the following observations.

- With narrow hysteresis margin, all scenarios run into several unnecessary hand-offs. This is the consequence of high variability and unreliability in mobile low-power links. A longer monitoring of stability m helps alleviating the ping-pong effects, however it enlarges the disconnection period.

- Thresholds at the higher end of the transitional region (-85 and -80 dBm) lead to a very long hand-off delay. This happens because mobile nodes tend to spend more time looking for overly reliable links and consequently less time transmitting data. The delivery ratio also follows the same fact by showing a decreasing trend from the lower end thresholds to the higher end.

- A wide hysteresis margin leads to the least number of hand-offs. Contrary to the narrower margin, monitoring the stability of the new AP for longer periods does not provide any further gain.

- The lower end threshold level maximizes the three metrics of interest.

- Utilizing a CSMA-based MAC demonstrates higher hand-off delay as the packet collision deteriorate the Discovery Phase.

- Entering two major patterns of interference namely periodic and sporadic with weak and strong transmit powers confirms the feasibility of smart-HOP. When interference is likely to occur, smart-HOP should utilize SNR-based parameters instead of RSSI.

They performed limited experiments in a controlled environment to discover the appropriate parameters for a hand-off process. Detailed information on the

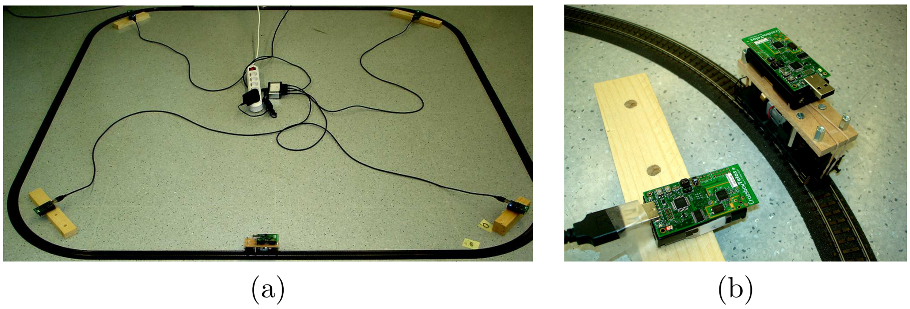

(a) (b)

Figure 4.2: *(a) 4 APs and a MN, (b) MN passing by an AP*

preliminary experiments is presented in [10]. A summary of the tests with the major findings are presented here.

Initial experimental setup. Calibrating the parameters of smart-HOP requires a testbed that provides a significant degree of repeatability. A fair comparison of different parameters is only possible if all of them observe similar channel conditions. In order to achieve this, a model-train was deployed in a large room. The room is 7 m×7 m and the locomotive follows a 3.5 m×3.5 m square layout. The speed of the locomotive was approximately 1 m/s (average walking speed). Figure 4.2(a) depicts a locomotive passing by an AP.

In real-world applications, the deployment of access points (or base stations) is subject to an accurate study to ensure the coverage of the area of interest. In cellular networks, the density of access points guarantees full coverage and redundancy. In other wireless networks, the density of access points depends on the real-time requirements of the application. In critical applications, such as the one considered in our report, complete coverage is an essential requirement. To prevent extreme deployment conditions such as very high or very low density of APs, our tests provide minimal overlap between contiguous APs. However, the distribution of access points is out of the scope of our report.

4.4 Thresholds, Hysteresis Margin and AP Stability

The first step in a hand-off scheme is to determine when should a node deem a link as weak and start looking for another AP. In our framework this is represented by TH_{low}. In the sensor networks community, the *de-facto* way to classify links is to use the connected, transitional and disconnected regions. In order to identify these regions, the RSSI and SNR values were gathered at different parts of the building utilizing different nodes. Figure 1.1 depicts these three regions for RSSI, which agree with previous studies [24]. The SNR parameters are used in the next section, when smart-HOP is evaluated under interference. The SNR is calculated by measuring the noise floor immediately after receiving the packet, and then, subtracting it from the RSSI value. The RSSI regions can be mapped directly to the SNR ones by subtracting the average noise floor.

An educated guess for the width of the hysteresis margin could be obtained from Figure 1.1 (based on the 10 dB width of the transitional region). However, while this value would guarantee that *all* links above TH_{high} are reliable, it would also increase the amount of beacons and time required to reach TH_{high}. In order to evaluate this region extensively, different values were considered for each hand-off parameter, as shown in Table 4.1. For example, if we consider scenario A with a 5 dBm margin and stability 2, it means that after the mobile node detects an AP above $TH_{high} = -90$ dBm, the node will send two 3-beacon bursts to observe if the link remains above TH_{high}. The hysteresis margin HM captures the sensitivity to ping-pong effects, and the number of bursts m, the stability of the AP candidate (recall that each burst in m contains three beacons).

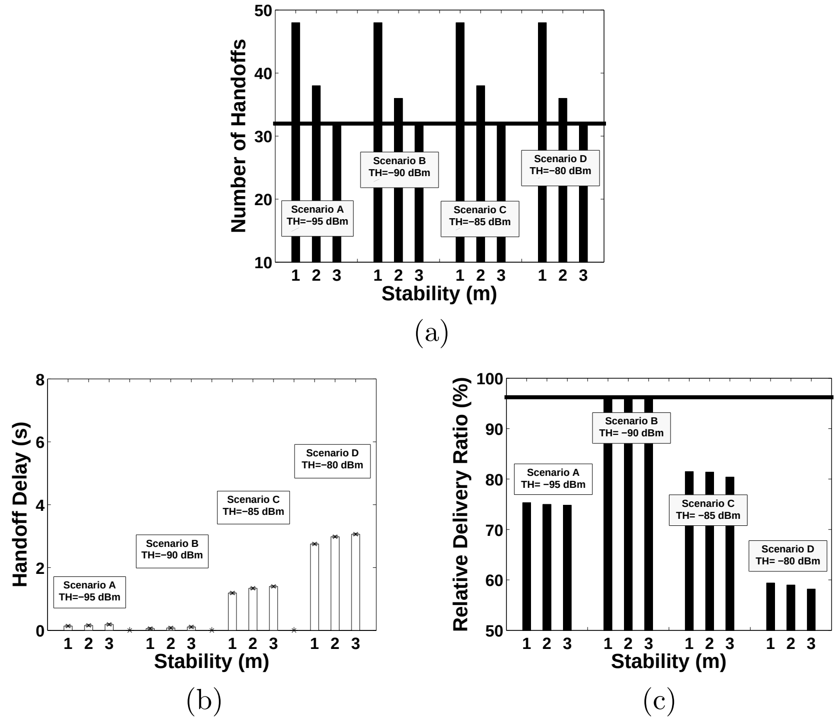
Table 4.1: *Description of second set of scenarios*

Scenarios	TH_{low}	HM	m
A	-95 dBm	1, 5 dBm	1, 2, 3
B	-90 dBm	1, 5 dBm	1, 2, 3
C	-85 dBm	1, 5 dBm	1, 2, 3
D	-80 dBm	1, 5 dBm	1, 2, 3

All scenarios of the experiment are shown in Table 1. The layout has four APs and one mobile node, as shown in Figure 4.2. For each evaluation tuple $< TH_{low}, HM, m >$, the mobile node takes four laps, which leads to a minimum of 16 hand-offs. The experiments provide some interesting results.

4.5 Observations

The high variability of low-power links can cause severe ping-pong effects. Figure 4.3(a) depicts the total number of hand-offs for the narrow margin case. We observe two important trends. First, all scenarios have ping-pong effects. The optimal number of hand-offs is 16, but all scenarios have between 32 and 48. Due to the link variability, the transition between neighboring APs requires between 2 and 3 hand-offs. Second, a longer monitoring of stability m helps alleviating ping-pong effects. We observe that for all scenarios the higher the stability, the lower the number of hand-offs.



(a)

(b)

(c)

Figure 4.3: *(a) number of hand-offs, (b) mean hand-off delay, (c) relative delivery ratio. The horizontal lines represent the results for the best scenario: 32 for the number of hand-offs and 96 for the relative delivery ratio. These values will be used as a reference in Figure 4.4.*

Thresholds at the higher end of the transitional region lead to longer delays and lower delivery rates. Figure 4.3(b) depicts the average hand-off delay for various thresholds TH_{low}. A threshold selected at the higher end of the transitional region (-85 or -80 dBm, scenarios C and D) can lead to an order of magnitude more delay than a threshold at the lower end (-90 dBm, scenario B). This happens because mobile nodes with higher thresholds spend more time looking for overly reliable links (more time on discovery phase), and consequently less time transmitting data (lower delivery rate). Figure 4.3(c) depicts the relative delivery rate and captures this trend. In order to have a reference for the absolute delivery rate, several broadcast scenarios were considered considering a high transmission rate and a 4-access point deployment. The average delivery rate was 98.2%, with a standard deviation of 8.7. This implies that there are limited segments with no coverage at all. Furthermore, the overlap is minimal which tests the agility of the hand-off mechanism (as opposed to dense deployments, where very good links are abundant). Scenario A in Figure 4.3(c) is an exception, because it remains disconnected for some periods of time. As shown in Figure 4(a), no link goes below -95 dBm, hence, when this threshold is used, the discovery phase does not start

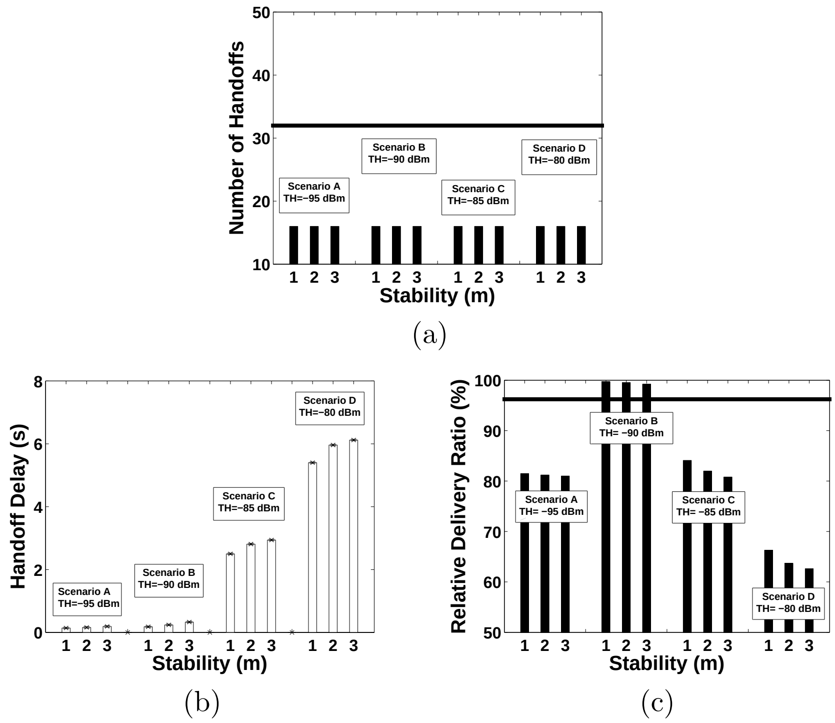


Figure 4.4: *(a) number of hand-offs, (b) mean hand-off delay, (c) relative delivery ratio. The horizontal lines represent the best results obtained for HM=1. The lines highlight the importance of an accurate calibration of the hand-off parameters.*

because the link goes below TH_{low}, but because disconnection time-outs occur.

The most efficient hand-offs seem to occur for thresholds at the lower end of the transitional region and a hysteresis margin of 5 dBm. Figure 4.4 shows that scenario B (-90 dBm) with stability 1 maximizes the three metrics of interest. It leads to the least number of hand-offs, with the lowest average delay and highest delivery rate. It is important to highlight the trends achieved by the wider hysteresis margin. First, the ping-pong effect is eliminated in all scenarios of Figure 4.4(a). Second, contrarily to the narrower hysteresis margin, monitoring the stability of the new AP for longer periods ($m = 2$ or 3) does not provide any further gains, because the wider margin copes with most of the link variability.

5

smart-HOP integration in RPL

5.1 Algorithm design

As mentioned on previous sections, smart-HOP involves two different phases (Data Transmission Phase and Discovery Phase). Algorithm 1 corresponds to the Data Transmission Phase, where the mobility detection is performed. The RSSI value is the main metric for smart-HOP decision, which is continuously collected. Since low-power links are usually asymmetric, the hand-off decision is based on RSSI readings at the AP side that are embed in the reply packets after receiving data messages. This is a key element to detect mobility in the network. By managing this continuous link quality observation, we can detect the exact moment of movement.

Algorithm 1: Data Transmission Phase

```
Mobility detection timer is implemented to monitor packet reception at the MN;
begin
    if reply packet is received then
        reset the timer;
        if $RSSI < TH_{low}$ then
            go to the Discovery Phase;
        else
            continue data communication;
        end
        reset hand-off timer;
    else
        if timer expires then
            MN unicasts burst of DIS messages to the serving parent;
            if DIO response is received then
                check DIO_RSSI value;
                if $DIO\_RSSI < TH_{low}$ then
                    parent in unreliable;
                    go to the Discovery Phase;
                else
                    parent is reliable;
                    continue data communication;
                end
            else
                go to the Discovery Phase;
            end
        else
            continue data communication;
        end
    end
end
```

A timer (*Mobility Detection Timer*) was implemented to detect the link degradation and parent unreachability. The current link is continuously assessed based on the average RSSI readings. If the RSSI value of the reply packet goes below a certain threshold, the MN enters the Discovery Phase. The MN should observe an activity during the timer period, otherwise it unicasts burst of DIS packets to the serving AP expecting to receive a DIO. Reception of good quality DIO reply allows the MN to continue data communication. If the RSSI level of the DIO message is not satisfactory, the hand-off process will be triggered. A silent parent —not responding by DIO message— is categorized as an unreachable parent, which leads to the Discovery Phase presented in Figure 5.1. This phase has three main steps as follows.

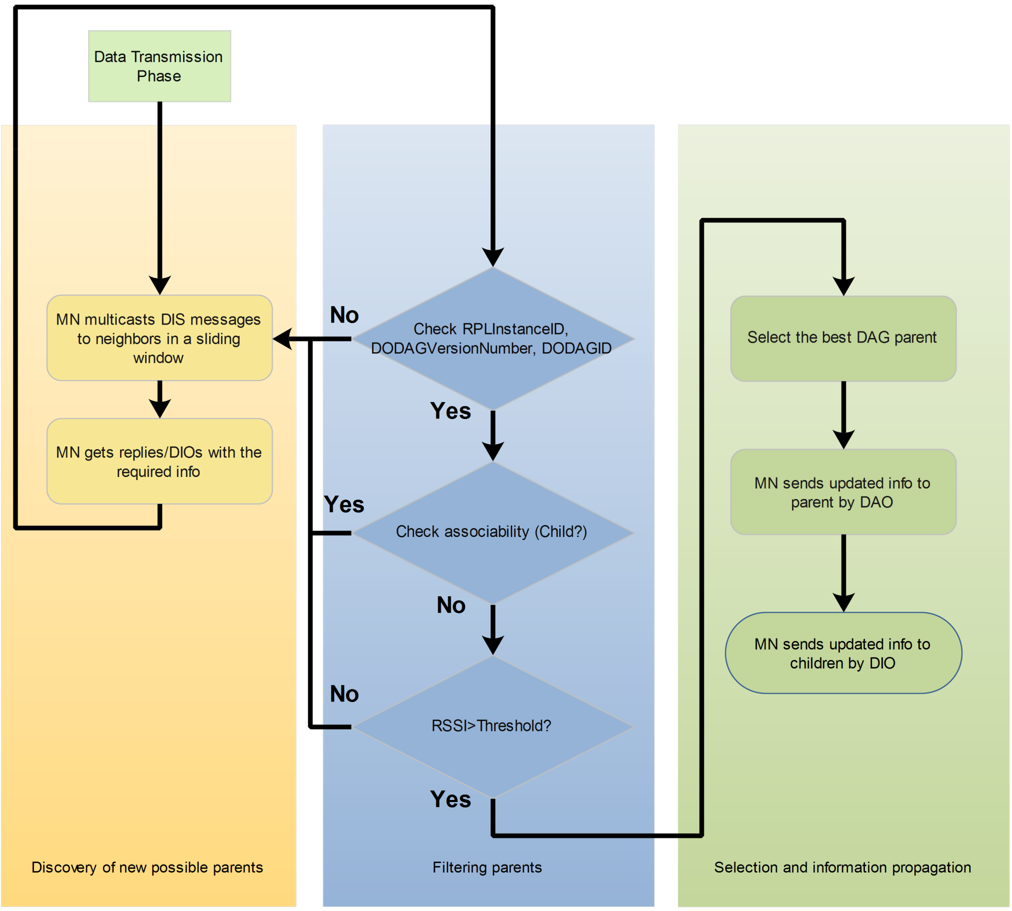

Data Transmission Phase
MN multicasts DIS messages to neighbors in a sliding window
MN gets replies/DIOs with the required info
Discovery of new possible parents
No
Check RPLInstanceID, DODAGVersionNumber, DODAGID
Yes
Yes
Check associability (Child?)
No
No
RSSI>Threshold?
Yes
Filtering parents
Select the best DAG parent
MN sends updated info to parent by DAO
MN sends updated info to children by DIO
Selection and information propagation

Figure 5.1: *Flowchart Discovery Phase*

- **Discovery of new possible parents**. MN multicasts DIS messages to neighbor nodes in a sliding window. By receiving the DIS packet at the APs, link quality metric is collected, averaged and embedded in a DIO unicast reply message.
- **Filtering parents**. After gathering the RSSI information from the neighbor nodes, the MN filters them according to a set of constraints. The *RPLIn-*

stanceID, *DODAGVersionNumber* and *DODAGID* are checked. Neighbors with unreliable link quality are discarded. This is the deciding factor to determine the best candidate.

- **Parent selection and information propagation**. After comparing link qualities, one parent with the highest and satisfactory quality is selected. This AP is assigned as the preferred parent. Then the MN sends a DAO message to the parent to create the downward route.

5.2 Implementation

To support smart-HOP within RPL routing, we applied several changes to enhance the routing process. The modifications do not interfere with the regular procedure of initial routing design. We describe the changes within RPL algorithm in four main classes: (i) trickle algorithm, (ii) control messages, (iii) timers, and (iv) medium access.

Trickle algorithm. According to the Trickle algorithm, every node broadcats messages (DIOs) to exchange information with local nodes. The interval of transmissions in bounded and enlarges if the network is stable. A mobile entity would interfere the network stability and hence the interval resets to the minimum value. To avoid this situation, we keep the Trickle interval unchanged during a hand-off process, instead the transmissions are scheduled independently.

Control messages: In order to handle the smart-HOP algorithm, we have enhanced the RPL control messages rather than creating new packets. The regular calling of each control message leads to a multicasting communication. In our design, we created more intelligent packets that obey both standard RPL routing and the smart-HOP process. It means that, these messages are transmitted on a regular basis, instead during the smart-HOP they follow specific rules.

In the Data Transmission Phase, the DIS is sent from MN to the AP (unicast) and the AP replies with a unicast DIO. The type of DIS and DIO is detected by reading a flag that reflects the status of each node. Figure 5.2 denotes the modifications made to the DIS base object.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|    Flags    | C | Reserved  |   Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5.2: *The modified DIS Base Object*

The first bit *F*, corresponds to the flag used to distinguish normal DIS messages from those triggered by the mobility process.

- Flag=0: represents standard RPL routing.
- Flag=1: represents Data Transmission Phase to verify parent reachability and in the Discovery Phase to look for possible parents.

The 2 bits following *Flags*, represent a counter *C*. This counter is responsible for identifying the multiple DIS messages triggered at the Discovery Phase. The maximum value of C is the efficient window size, which is set by application user. In smart-HOP ws=3 would lead to C=3 in RPL algorithm. If a DIS is being sent in the Data Transmission Phase, this counter contains a 0 value. Hence, we use this counter to distinguish a DIS between the two phases.

According to the specification, DIS and DIO messages have the bytes Flags and Reserved equals to 0. Similarly, the DIO message was also modified (Figure 5.3)in order to include this flag, but here, it has 2 bits.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |Version Number |             Rank              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|0| MOP | Prf |     DTSN      | F |   Flags   |      RSSI     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID                            +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 5.3: *The modified DIO Base Object*

When a parent sends a DIS with a flag, a DIO response is expected, and this DIO needs to carry a flag so that standard/periodic DIOs don't trigger an unexpected behavior throughout a mobility process. The reason why DIO needs 2 bits for the flag, is that there are three possible occurrences:

- Flag=0: for standard RPL DIO transmission.

- Flag=1: for DIO reply in Data Transmission Phase when parent link quality is being assessed.
- Flag=2: for DIO reply in Discovery Phase when multiple DIS messages are received.

Reserved is being used to accommodate the RSSI that is read by the parent upon DIS reception. As stated earlier, links are asymmetric, hence the need to read the link quality and send it back to the mobile node.

Timers. We have implemented three main timers to support mobility.

(i) *Mobility Detection Timer.* We run a timer on all nodes with mobility feature that increases RPL routing responsiveness drastically. This timer is set to a value, which provides the application user requirements. During this period, the MN keeps listening to the channel to monitor the incoming packets from the serving parent. By elapsing the timer period if the MN observes a silent parent, then it resumes a Discovery Phase. In case of getting replies from the parent (e.g. Trickle DIO, unicast DIO or a data packet), the timer is reset. Figure 5.4 represents the Data Transmission Phase.

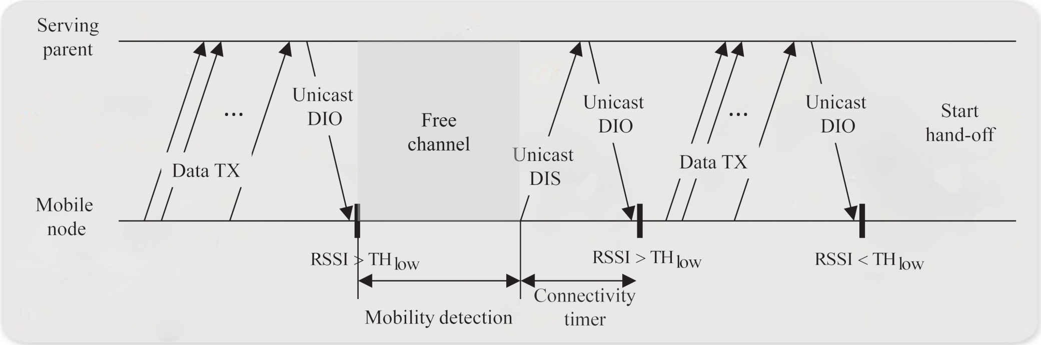


Figure 5.4: *Timing diagram of Data Transmission Phase.*

(ii) *Connectivity Timer.* This timer is started when the mobile node is assessing the link quality of its current parent. Upon sending the DIS, the MN needs to wait for a determined period that corresponds to the expected time it takes for a DIO reply to be received. If in a meanwhile, a DIO is received, the quality is assessed and MN acts accordingly to the RSSI value. If it is above a certain threshold level, communication continues, if not, the hand-off process is started. If the timer expires, it means that DIO was not received and the parent is considered as unreachable.

(iii) *Hand-off Timer.* As depicted in Figure 5.5, this timer comprises all the packet exchanges within a hand-off process. This phase starts by sending burst of DIS messages to the neighboring nodes in a sliding window. It is important to calculate the maximum possible rate of DIS transmission in order to guarantee the successful reception of these control messages at the AP side and to process this message according to the regulations imposed by RPL and smart-HOP processes. We implemented a backoff timer to manage the DIS transmissions.

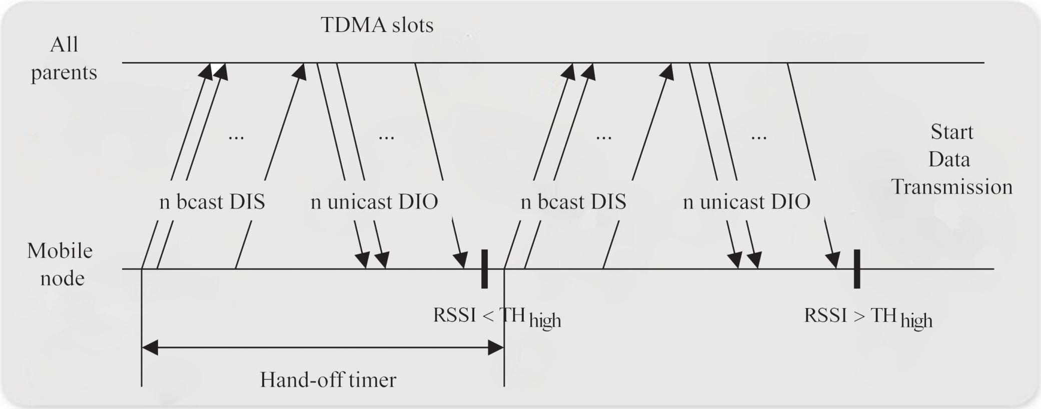


Figure 5.5: *Timing diagram of Discovery Phase.*

The neighbor parents are supposed to reply after receiving burst of DIS, a number of DIS messages equal to the window size. Finding a proper moment for replying is very important as the MN may not be ready to get the reply and it may interfere with other reply packets sent from neighbors. The low-power and lossy links are not very stable to expect reception of all DIS messages at the AP side. Hence, we implemented another timer that distinguishes the waiting time ,before reply, according to (i) the information extracted from the DIS and (ii) the DIS reception time. The main information of the DIS message is the C that shows the sequence number of the DIS message. This timer is tuned according to the sequence information, which obliges the AP to wait for DIO transmission for certain amount of time. Figure 5.6 represents the mechanism of this self scalable timer.

Reminding that all the messages are tagged with a *Counter* field —see Figure 5.2, the AP is able to predict the reception moment of all DIS messages, which is calculated as follows: $(ws - C) * T\_DIS$. T_DIS corresponds to the maximum DIS interval, set at the MN. As depicted in Figure 5.6, there are three possible situations (assuming ws=3).

- A: corresponds to the reception of the first DIS with the field *Counter* equals to 1. Using the expression $(3 - 1) * T\_DIS$, the timer will be equivalent to the reception of two additional DIS messages. Considering (once again) the lossy

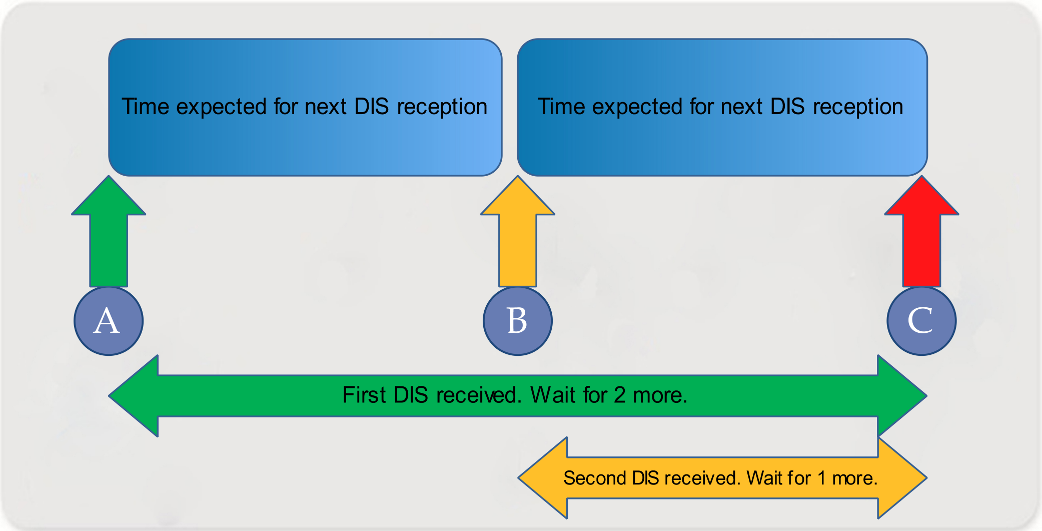

Time expected for next DIS reception
Time expected for next DIS reception
A
B
C
First DIS received. Wait for 2 more.
Second DIS received. Wait for 1 more.

Figure 5.6: *DIS reception mechanism - self scalable timer*

nature of the link, we might not receive the second message, hence the reason to wait twice the normal period.

- B: corresponds to the reception of the second DIS with the field *Counter* equals to 2. Using the expression, the timer will start with the equivalent to one DIS reception.

- C: corresponds to the reception of the third and last DIS message with the field *Counter* equals to 3. Using the expression, no timer will be started and the preparation of the DIO reply message is started.

MN may receive one or more replies according to the link quality. According to the standard RPL algorithm, a node is supposed to process DIO transmission immediately after DIS reception. smart-HOP imposes some additional regulations to follow the *Hand-off Timer*. A major share of the hand-off delay is due to this timer. The DIO reply is sent with *Flag* = 2 (see Figure 5.3), so the MN can distinguish it from standard DIOs and act accordingly. To schedule the sequence of sending these replies, each parent runs a medium access algorithm that reduces the possibility of collision.

Medium access. In our implementation, we use motes with 802.15.4 radio (250 Kbit/s). The packet size depends on the data payload, which is added to the header and footer. Since RPL runs an IPv6 addressing strategy, we assume that the packet size is 127 bytes in worst case. Considering the radio data rate and the packet size, the mote is able to transmit at most 246 packets/s (≈4 ms). The propagation delay, modulation, demodulation, fragmentation and de-fragmentation extends this approximate transmission delay. It is wise to select intervals in the range of 10 to 15 ms to ensure successful transmissions.

Based on the link quality level, each parent decides to advertise its reading or to remain silent. Since parents with poor link quality ($RSSI < TH_{high}$) are excluded from the possible parent set of the MN, thus, these readings are not advertised to reduce the collision. Each parent assigns a priority to the acceptable range of RSSI reading as shown in Table 5.1. The priorities are used in scheduling the DIO transmissions in different slots. Since the low-power networks are mostly working in the transitional region, it is more probable that different parents choose same slot.

Table 5.1: *Priority assignment*

Priority	Range of average RSSI reading
0	$-85 < RSSI < -80$ dBm
1	$RSSI \geq -80$ dBm

The DIOs information is saved and associated with the corresponding parent address. After the hand-off timer ends, this information is compared in order to attain the best possible parent. As stated in the previous section, the best parent is defined as preferred parent and a route is added, concluding the mobility process.

5.3 Evaluation

To perform an evaluation of smart-HOP, Cooja simulator was used together with a mobility plugin. This plugin allows nodes to move according to a model. We used 2 different models to gather statistic data and compare smart-HOP with standard RPL. The first model was used to attain a preliminary result on the amount of control messages needed and hand-off delay. The second model was used to execute a more thorough evaluation of smart-HOP performance during mobility. 5.7 represents the first model used.

All nodes start at the same position and remain there for 10 seconds. Node 1 corresponds to the MN, 4 and 5 are the possible parents. After 10 seconds, MN starts moving at a rate of 2 meters/sec to the vicinity of 5 and remains there until the simulation ends. MN sends packets every 20 ms for a simulation time of 12 minutes (assured by the simulation script used).

As stated previously, current RPL behavior relies on DIO messages to find new parents. With this in mind, smart-HOP was compared with 3 different scenarios of standard RPL:

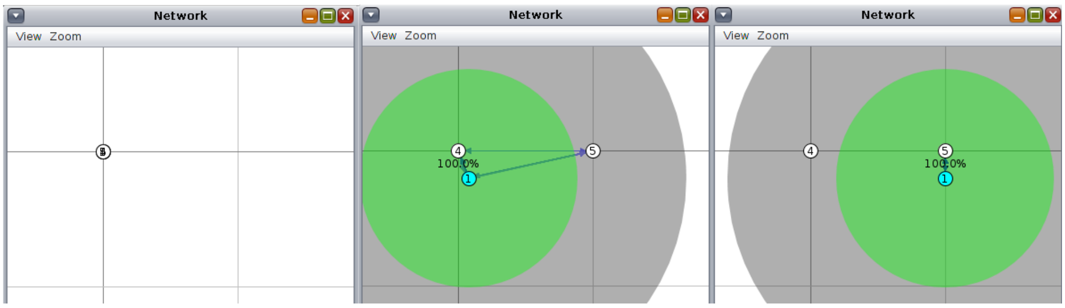


Figure 5.7: *Simulation topology*

- Standard RPL: This is de default configuration in Contiki with $RPL\_DIO\_INTERVAL\_MIN = 12$ and $RPL\_DIO\_INTERVAL\_DOUBLINGS = 8$. This corresponds to a minimum DIO delay of approximately 4 seconds, scaling to a maximum of 17min. (refer to 3.6)

- Standard RPL 2: This configuration uses $RPL\_DIO\_INTERVAL\_MIN = 8$ and $RPL\_DIO\_INTERVAL\_DOUBLINGS = 1$. This corresponds to a minimum DIO delay of 256 ms, scaling to a maximum of 512 ms.

- Standard RPL 3: This configuration uses $RPL\_DIO\_INTERVAL\_MIN = 7$ and $RPL\_DIO\_INTERVAL\_DOUBLINGS = 1$. This corresponds to a minimum DIO delay of 128 ms, scaling to a maximum of 256 ms.

The comparison with the modified RPL configurations were performed to try and match the mobility delay achieved by smart-HOP. The results are presented in 5.8

Standard RPL 3 is not represented because there is no actual value for this simulation. The DODAG is so unstable that the MN cannot choose one fixed parent and keeps sending packets in an alternating sequence. This is because he receives DIOs at such a high rate, he can not pick one parent to stabilize. With this scenario, there is no disconnection of the MN and "mobility" is performed even before the MN has a bad link with previous parent. The amount of control packets to achieve this is unsustainable and even the MN random routing behaviour would affect the overall network performance.

The performance results represented in 5.8, show that smart-HOP achieved a hand-off delay of only 85 ms, reducing significantly the time required for RPL to find a new parent. The results shown in 5.9 represent a comparison of the control packets needed by each scenario.

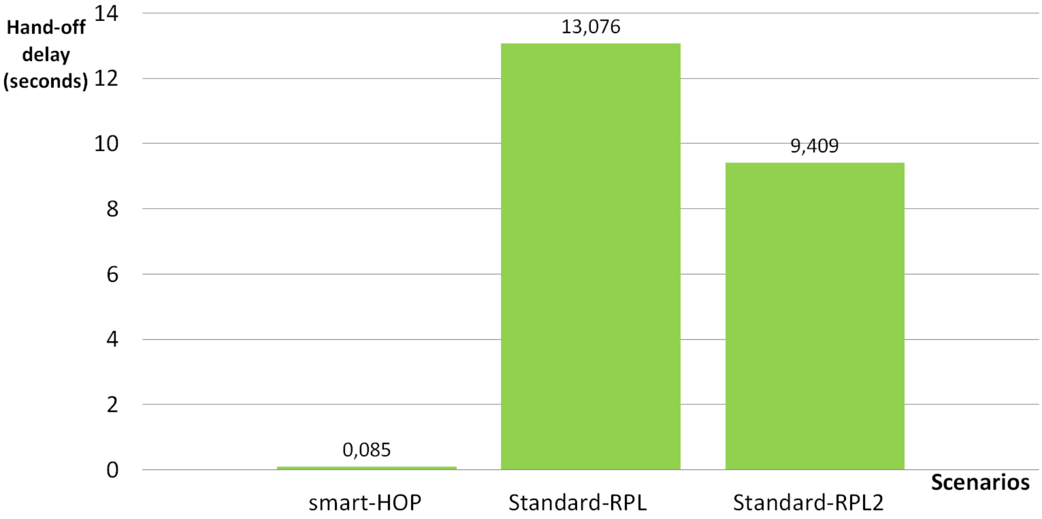


Figure 5.8: *Mobility delay comparison between smart-HOP and RPL scenarios*

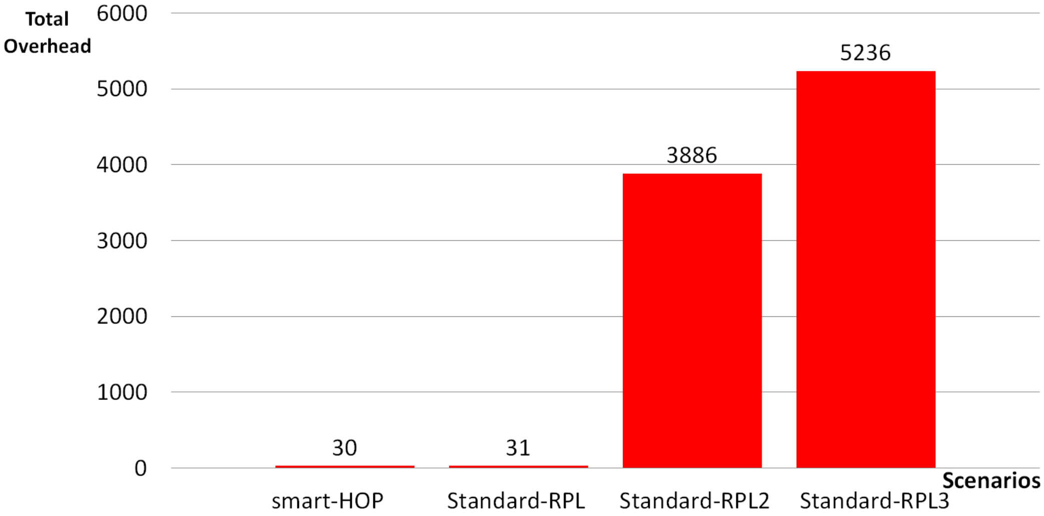


Figure 5.9: *Model Comparison - Packet Statistics*

Note that smart-HOP is supported by standard RPL periodic sending of DIO messages, hence the 30 packets of payload. The increasing rate of DIO messages to match smart-HOP hand-off delay performance (Standard RPL 3) is not worth the amount of stress imputed to the network. This type of usage is simply not sustainable, increasing significantly the power consumption on the devices.

Instead of relying in just one measure of the smart-HOP hand-off delay, a second simulation was performed in order to assess this value with higher precision. 5.10

shows the topology used for this test.

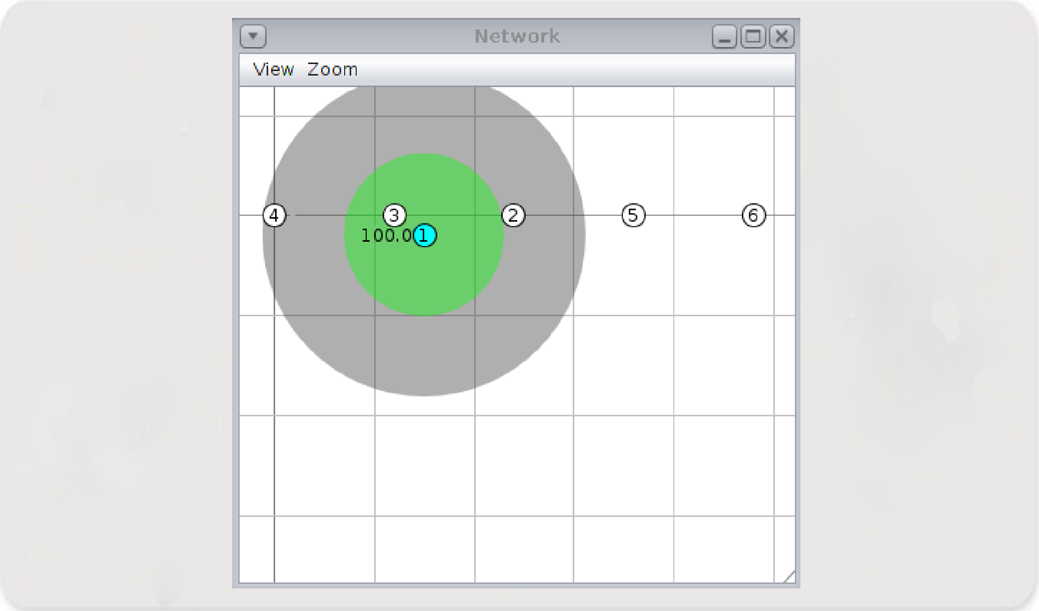


Figure 5.10: *smart-HOP thorough simulation*

The simulation configuration is similar to the one used in 5.7. The only change is the number of nodes, allowing consecutive hand-offs, providing this way a better assessment of the smart-HOP performance. Twenty hand-offs were carried through with an average of 85.75 ms hand-off delay. Confirming this way the robustness of smart-hop towards consecutive parent changes.

6 Conclusions

Many critical low-power wireless network applications need not only reliability, but also the ability to adequately cope with the movement of nodes. smart-HOP solves the problem of delivering data from a mobile node to a static point of attachment implementing an efficient hand-off process. smart-HOP is a hand-off process tailored for wireless sensor networks. This algorithm has two main phases: *(i) Data Transmission Phase* and *(ii) Discovery Phase.* The mobile node starts the Discovery Phase when the link quality goes below a certain threshold (TH_{low}) and looks for APs that are above a reliable threshold ($TH_{high} = TH_{low} + HM$, where HM is the hysteresis margin). The most efficient hand-offs seem to occur for thresholds at the lower end of the transitional region and a hysteresis margin of 5 dBm.

To support smart-HOP within RPL routing, we applied several changes to enhance the routing process. The modifications do not interfere with the regular procedure of initial routing design. In our design, we created more intelligent packets that obey both standard RPL routing and the smart-HOP process. Implemented new timers to assess the disconnection of nodes, wait for neighbors information and to avoid collision while replies occur.

We proposed an algorithm to integrate smart-HOP within RPL routing protocol. The main issue in this implementation is the reuse of existing control messages to support a reliable mobility support. The Data Transmission Phase is enriched by a periodic link assessment process by exchanging unicast DIO and DIS messages. The Discovery Phase is instead initiated by multicasting DIS messages to explore neighbor nodes. A filtering process is then applied to select a parent with the highest link quality level, while avoids loop after hand-off process.

smart-HOP was also integrated within the Trickle algorithm running on top of RPL routing protocol. The regular data communication and neighbor discovery follow the adaptive DIO timing behavior, while smart-HOP allows periodic signaling at certain moments without interfering the Trickle periods.

The evaluation results show that our proposed scheme minimizes the signaling cost and the hand-off delay. smart-HOP performs a hand-off process within an average period of 85 ms, outperforming the standard RPL delay of approximately 614 seconds. The simulation results indicate the reliability of smart-HOP for multi-hop standard technologies like RPL. However, it is important to perform real experiments with motes to further analyze and fine tune the relevant parameters.

Bibliography

[1] Z. Shelby and C. Bormann, *6lowpan: the wireless embedded internet* (Wiley, 2009).

[2] J. Hui and P. Thubert, “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks,”, RFC 6282 (Proposed Standard), 2011.

[3] O. Gaddour and A. KoubíA, “Survey RPL in a nutshell: A survey,” Comput. Netw. **56,** 3163–3178 (2012).

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,”, RFC 6550 (Proposed Standard), 2012.

[5] A. de Pablo Escolà, “Development of a wireless sensor network with 6LoWPAN support,” (2009).

[6] H. Alemdar and C. Ersoy, “Wireless sensor networks for healthcare: A survey,” Computer Networks **54,** 2688–2710 (2010).

[7] H. Qin, Z. Li, Y. Wang, X. Lu, W. Zhang, and G. Wang, “An integrated network of roadside sensors and vehicles for driving safety: Concept, design and experiments,” In *IEEE PERCOM 2010*, (2010).

[8] B. C. Villaverde, S. Rea, and D. Pesch, “InRout - A QoS aware route selection algorithm for industrial wireless sensor networks,” Ad Hoc Networks (2011).

[9] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman, “Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit,” In , SenSys 2010 (2010).

[10] H. Fotouhi, M. Zuniga, M. Alves, A. Koubaa, and P. Marron, “Smart-HOP: A Reliable Handoff Mechanism for Mobile Wireless Sensor Networks,” in *EWSN Conference*, Vol. 7158 of *Lecture Notes in Computer Science*, G. Picco and W. Heinzelman, eds., (Springer Berlin / Heidelberg, 2012), pp. 131–146.

[11] N. Baccour, A. Koubâa, H. Youssef, M. Ben Jamâa, D. do Rosário, M. Alves, and L. Becker, “F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks,” in *EWSN 2010* (2010).

[12] K. Srinivasan and P. Levis, “RSSI is Under Appreciated,” In *EmNets 2006*, (2006).

[13] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," In *Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI)*, 2007 (2007).

[14] C. Boano, T. Voigt, A. Dunkels, F. Osterlind, N. Tsiftes, L. Mottola, and P. Suarez, "Exploiting the LQI variance for rapid channel quality assessment," In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pp. 369–370 (2009).

[15] S. Cho, E. Jang, and J. Cioffi, "Handover in multihop cellular networks," Communications Magazine, IEEE **47,** 64 –73 (2009).

[16] D. Wong and T. J. Lim, "Soft handoffs in CDMA mobile systems," Personal Communications, IEEE 4 (1997).

[17] Y. Ma, J. Han, and K. Trivedi, "Call admission control for reducing dropped calls in code division multiple access (CDMA) cellular systems," In *INFOCOM 2000*, (2000).

[18] T. Salih and K. Fidanboylu, "Modeling and analysis of queuing handoff calls in single and two-tier cellular networks," Computer Communications 29 (2006).

[19] B. Madan, S. Dharmaraja, and K. Trivedi, "Combined Guard Channel and Mobile-Assisted Handoff for Cellular Networks," Vehicular Technology, IEEE Transactions on **57,** 502 –510 (2008).

[20] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," In *INFOCOM 2005*, (2005).

[21] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," SIGCOMM 2003 (2003).

[22] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graphs," In , MobiSys 2004 pp. 70–83 (2004).

[23] M. Zuniga, I. Irzynska, J. Hauer, T. Voigt, C. Boano, and K. Roemer, "Link quality ranking: Getting the best out of unreliable links," In *DCOSS 2011*, (2011).

[24] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The b-factor: measuring wireless link burstiness," In *ACM SenSys 2008*, (2008).

[25] X. Wang, S. Xie, and X. Hu, "Recursive analysis for soft handoff schemes in CDMA cellular systems," IEEE Trans. Wireless Communications, 8 (2009).

[26] A. Hasler, I. Talzi, C. Tschudin, and S. Gruber, "Wireless sensor networks in permafrost research - concept, requirements, implementation and challenges," In *Proc. 9th Int'l Conf. on Permafrost (NICOP 2008*, (2008).

[27] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," (2004).

[28] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-based communication in wireless sensor networks," IEEE Communications Magazine **44,** 56 (2006).

[29] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," In *Information Processing in Sensor Networks*, pp. 552–552 (2003).

[30] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," Ad Hoc Networks **1,** 215–233 (2003).

[31] J. she Jin, J. Jin, Y. hui Wang, K. Zhao, and J. jun Hu, "Development of Remote-Controlled Home Automation System with Wireless Sensor Network," In *Embedded Computing, 2008. SEC '08. Fifth IEEE International Symposium on*, pp. 169–173 (2008).

[32] J. A. Gutierrez, E. H. Callaway, and R. L. Barrett, *Low-rate wireless personal area networks: enabling wireless sensors with IEEE 802.15. 4* (Institute of Electrical & Electronics Engineers (IEEE), 2004).

[33] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Comput. Netw. **52,** 2292–2330 (2008).

[34] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo, "The emergence of a networking primitive in wireless sensor networks," Commun. ACM **51,** 99–106 (2008).

[35] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4-2003 pp. – (2003).

[36] R. N. M. da Silva, "Service Discovery and Mobility in Real Scenarios," (2008).

[37] S. Kent and K. Seo, "Security Architecture for the Internet Protocol,", RFC 4301 (Proposed Standard), 2005, updated by RFC 6040.

[38] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification,", RFC 2460 (Draft Standard), 1998, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946.

[39] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,", RFC 4919 (Informational), 2007.

[40] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),", RFC 6775 (Proposed Standard), 2012.

[41] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, “IPv6 Addressing of IPv4/IPv6 Translators,”, RFC 6052 (Proposed Standard), 2010.

[42] S. Thomson, T. Narten, and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,”, RFC 4862 (Draft Standard), 2007.

[43] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on,* pp. 455–462 (2004).

[44] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with cooja,” In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on,* pp. 641–648 (2006).

[45] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer networks **38,** 393–422 (2002).

[46] R. Estanjini, Y. Lin, K. Li, D. Guo, and I. Paschalidis, “Optimizing warehouse forklift dispatching using a sensor network and stochastic learning,” Industrial Informatics, IEEE Transactions on **7,** 476–486 (2011).

[47] V. Gungor and G. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” Industrial Electronics, IEEE Transactions on **56,** 4258–4265 (2009).

[48] D. Shah and S. Shakkottai, “Oblivious routing with mobile fusion centers over a sensor network,” In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE,* pp. 1541–1549 (2007).

[49] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, “Mobeyes: smart mobs for urban monitoring with a vehicular sensor network,” Wireless Communications, IEEE **13,** 52–57 (2006).

[50] T. Brennan, L. Leape, N. Laird, L. Hebert, A. Localio, A. Lawthers, J. Newhouse, P. Weiler, and H. Hiatt, “Incidence of adverse events and negligence in hospitalized patients,” New England journal of medicine **324,** 370–376 (1991).

[51] A. Initiative, “National Patient Safety Goals,” Hospital Pharmacy **38,** 490–496 (2008).

[52] R. Wilson *et al.*, “The quality in Australian health care study,” Medical Journal of Australia **163,** 458–471 (1995).

[53] L. Leape, T. Brennan, N. Laird, A. Lawthers, A. Localio, B. Barnes, L. Hebert, J. Newhouse, P. Weiler, and H. Hiatt, “The nature of adverse events in hospitalized patients,” New England Journal of Medicine **324,** 377–384 (1991).

[54] S. Pack, H. Jung, T. Kwon, and Y. Choi, “SNC: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks,” SIGMOBILE Mob. Comput. Commun. Rev. **9,** 39–49 (2005).

[55] J. Petajajarvi and H. Karvonen, "Soft handover method for mobile wireless sensor networks based on 6LoWPAN," In *IEEE DCOSS 2011*, (2011).

[56] Z. Zinonos and V. Vassiliou, "S-GinMob: Soft-handoff solution for mobile users in industrial environments," In *IEEE DCOSS 2011*, pp. 1 –6 (2011).

[57] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," In *MobiCom 2003*, (2003).

[58] J. Tripathi, J. De Oliveira, and J. P. Vasseur, "A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks," In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pp. 1–6 (2010).

[59] J. Tripathi, J. de Oliveira, and J. Vasseur, "Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL),", RFC 6687 (Informational), 2012.

[60] M. R. Khan, Ph.D. thesis, KTH, 2012.

[61] H. Ali, "A Performance Evaluation of RPL in Contiki," .

[62] V. M. Bauer, "Routing in Wireless Sensor Networks: An Experimental Evaluation of RPL," .

[63] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo, "The emergence of a networking primitive in wireless sensor networks," Commun. ACM **51,** 99–106 (2008).

[64] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm,", RFC 6206 (Proposed Standard), 2011.

[65] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for sensor networks," In *in Ambient Intelligence*, (Springer Verlag, 2004).

[66] *TelosB Datasheet* (Crossbow, 2013).